VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Aplikace metody business inteligence pro transformaci dat do datového skladu

Application of the Method of Business Intelligence to Transform Data to Data Warehouse

Student: Lukáš Derján
Vedoucí bakalářské práce: doc. Ing. Milena Tvrdíková

Ostrava 2013

VŠB - Technická univerzita Ostrava
Ekonomická fakulta
Katedra aplikované informatiky

Zadání bakalářské práce

Student: **Lukáš Derján**

Studijní program: B6209 Systémové inženýrství a informatika

Studijní obor: 6209R001 Aplikovaná informatika

Téma: Aplikace metody business intelligence pro transformaci dat do datového skladu
Application of the Method of Business Intelligence to Transform Data to Data Warehouse

Zásady pro vypracování:

1. Úvod
2. Teoretická východiska a infrastruktura business intelligence
3. Analýza využití technologie ETL v dané oblasti
4. Návrh a praktická realizace aplikace pro transformaci dat do datového skladu
5. Zhodnocení řešení
6. Závěr

Seznam použité literatury
Seznam zkratek
Prohlášení o využití výsledků bakalářské práce
Seznam příloh
Přílohy

Seznam doporučené odborné literatury:

INMON, William H. *Building the Data Warehouse*. 4th ed. Indianapolis: Wiley, 2005. ISBN 978-0-7645-9944-6.
KIMBALL, Ralph a Joe CASERTA. *The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data*. Indianapolis: Wiley, 2004. ISBN 07-645-6757-8.
LACKO, Luboslav. *Databáze: datové sklady, analýza OLAP a dolování dat*. Brno: Computer Press, 2003. ISBN 80-7226-969-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Milena Tvrdíková, CSc.**

Datum zadání: 23.11.2012
Datum odevzdání: 10.05.2013

VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA Ekonomická fakulta OSTRAVA 1

Ing. Petr Rozehnal, Ph.D.
vedoucí katedry

prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Prohlášení

„Místopřísežně prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně.“

.....

Lukáš Derján

Poděkování

Touto formou bych rád poděkoval doc. Ing. Mileně Tvrdíkové, CSc. za poskytnutí mnoha podnětných rad a také kolegům ze společnosti Tieto, jmenovitě Mgr. Janu Hřivňákovi za odborné vedení a Tomáši Holblingovi za svěření projektu a poskytnuté podklady, které mi pomohly k vypracování této práce.

Obsah

1 Úvod

Téměř s jistotou je možné tvrdit, že v dnešní době není firmy, korporace, menší společnosti nebo drobného podnikatele, který by nějakým způsobem nevyužíval databázové systémy, či již v dnešní době ojedinělé souborové, neboli agendové, zpracování dat. Ať už pro potřebu archivace zakázek, tvorby jednoduchého seznamu zákazníků či přehledy svých prodejů.

Pro úspěšnost při konkurenčním boji je však třeba tyto údaje správně využít. Větší firmy se potýkají s vysokou kumulací dat různorodého charakteru, jež se s přibývajícím časem a rostoucím množstvím stávají složitější pro zpracování a ztrácí tak svou vypovídající hodnotu. Problém pak nastává v situacích, kdy je potřeba data analyzovat v kratším časovém úseku.

Při operativním rozhodování je důležité mít k dispozici vypovídající údaje v co nejkratším možném čase. Platí zde lidové rčení, že čas jsou peníze. Zde se vedení jisté společnosti či menšího podniku nabízí možnost zvážit využití business intelligence jakožto procesu, který nabízí řešení v podobě datových skladů a analýzy nad nimi, kdy úspora peněz i času díky těmto analýzám není zanedbatelná.

Samotný pojem business intelligence by se dal popsat jako proces přeměny údajů ve vypovídající informace pro koncového zákazníka. Zákazník je v této práci chápan jako společnost, jež má zájem zefektivnit rozhodování na všech úrovních řízení.

Infrastrukturou business intelligence řešení jsou datové sklady. Data neboli údaje, v datovém skladu obsažená, je třeba uchovávat v jisté formě, s určitou hodnotou, která se projeví převážně až při následném využití těchto dat pro potřeby reportingu či analýz.

Cílem této bakalářské práce je nejen průběh procesu přesunu dat ze zdrojových systémů zákazníka do datového skladu navrhnout, ale i prakticky realizovat v reálném projektu.

První část je popisem teoretických východisek business intelligence a technologie datových pump. Proces přesunu dat totiž není jen pouhým kopírováním dat zákazníka do datového skladu, jak by mohl být mylně vykládán. Jedná se o komplexní postup, kdy jsou data extrahována, transformována a následně nahrána do předem připravených struktur datového skladu. Právě tyto zmíněné operace (extrakce, transformace a nahrání) jsou

i s návrhem a praktickou realizací označovány souhrnně jako datová pumpa. Z těchto tří operací je pak odvozena zkratka názvu tohoto procesu jakožto ETL.

Druhá část této bakalářské práce se zabývá popisem a analýzou reálného projektu, pro který je proces navrhován, se stručnou charakteristikou společnosti, jež byla dodavatelem koncového řešení a v rámci které byla tato práce vypracována.

Následuje samotný návrh a praktická realizace řešení pro koncového zákazníka.

2 Teoretická východiska a infrastruktura business inteligence

Cílem následující kapitoly je definovat základní pojmy z oblasti business inteligence a seznámit čtenáře s problematikou datových skladů.

Existuje mnoho definicí business intelligence (BI) jakožto celku, všechny se však shodují v základním popisu BI, jakožto skupiny nástrojů a úkonů, jejichž cílem je získávání informací z ukládaných dat, která pak dále slouží pro podporu strategického a operativního rozhodování.

"The processes, technologies and tools needed to turn data into information and information into knowledge and knowledge into plans that drive profitable business action. BI encompasses data warehousing, business analytics and knowledge management."

(Loshin, 2003, s. 6)

Definice se zmiňuje o datech, informacích, znalostech a datových skladech. Velmi obecně lze BI popsat jako proces přeměny dat v informace a informace ve znalosti (1). Tyto pojmy jsou pro tuto bakalářskou práci určující a budou v následujícím textu přesně definovány.

BI ovšem není jen o výstupech, vyskytujících se nejčastěji ve formě reportů či dashboardů, jež zákazníkovi poskytují data transformovaná ve znalosti. BI nejsou jen kombinací programových nástrojů a dotazovacích jazyků. Komplexní BI systém umožní společnosti nejen určit statistické údaje, jako nejprodávanější zboží, či lokality s nejvěrnějšími zákazníky.

Starting a well-conceived and comprehensive BI practice will not just provide the physical tools for answering these kinds of questions, but, more importantly, should be a catalyst for a change in the way we think about doing business and about how we can use information within that new way of thinking.

(Loshin, 2003, s. 3)

BI umožní předem identifikovat kdo je dobrý a kdo špatný zákazník. Co je pro společnost ono *dobro* či *zlo*. V rámci BI jsou vytvořeny metriky, jež jsou využívány právě k určení a rozlišení těchto typů zákazníků. Je definováno, jaká data musí být shromažďována pro taková měření a je zajištěna kvalita těchto údajů tak, aby nedošlo k vyvozování mylných

závěrů (1). Způsob prezentování výsledků je rychlý (i real-time) a efektivní, sloužící pro podporu kvality rozhodování.

Michal Hroch s Pavlem Cachem, autoři článků online magazínu Systém OnLine, definují BI jako souhrnný pojem pro procesy, technologie a nástroje potřebné k přetvoření dat do informací, informací do znalostí a znalostí do plánů, které umožní provést akce podporující splnění primárních cílů organizace (2).

Význam slov data či údaje je třeba chápat v jednom a tom samém kontextu, neb z údajů ještě nemusí zcela jasně vyplývat samotná informace. Data se stávají informacemi až tehdy, mají-li pro uživatele konkrétní význam. Analogicky jsou si významem shodná slova informace a poznatky. Znalosti jsou pak v daném problému aplikované informace. Jsou tedy tím, co jedinec ví po osvojení dat a začlenění informací v souvislostech. V definicích jsou pak tyto pojmy vyjádřeny jako:

- ***Data*** - *„Účelem dat je přenášet a dále zpracovávat odraz skutečnosti. Jsou to jakékoli zaznamenané poznatky či fakta." (3)*,

- ***Informace*** - *„Informace je to, co vede k odstranění existující neurčitosti, nejasnosti nebo nevědomosti." (4)*,

- ***Znalosti*** - *„Znalost obsahuje pravdy a přesvědčení, perspektivy a koncepty, úsudky a očekávání, metodologie a know-how." (5)*.

Datové sklady, jež podle názvu slouží pro skladování již zmiňovaných dat, je možno rozdělit dle použitého typu přístupu při jejich budování. Přístup integrovaný a přístup dimenzionální. Každý z těchto přístupů je prosazován rozdílným autorem a v průběhu následující kapitoly bude přiblížen z pohledu obou.

„The users of an operational system turn the wheels of the organization.(...) The users of a data warehouse, on the other hand, watch the wheels of the organization turn."

(Kimball, 2002, s. 2)

Citace pochází z významného díla Ralpha Kimballa *The Data Warehouse Toolkit* (dnes již ve druhém vydání). Autor jednoznačně, přesto s nadhledem a jednoduchostí zdůrazňuje důležitost datových skladů v rámci BI (6). Kniha samotná je již od svého prvního vydání z roku 1996 fundamentální záležitostí dimenzionálního přístupu budování datových

skladů, jež Kimball prosazuje. S nadsázkou lze knihu označit jako povinnou četbu každého, kdo má zájem o práci s datovými sklady či v oboru BI.

Druhým velmi významným autorem v rámci budování datových skladů je Bill Inmon. Ten ve své knize *Building The Data Warehouse,* jež vyšla v roce 2005 již ve 4. vydání, prosazuje přístup *integrovaný*. Právě s Inmonem je spojován titul otce datových skladů. Prosazuje integrovaný přístup, také nazýván jako relační (7) či normalizovaný a je stavěn do pozice přímého konkurenta přístupu dimenzionálního.

Problematika integrovaného i dimenzionálního přístupů bude podrobněji rozebrána v průběhu bakalářské práce (podkapitola 2.2).

BI, v nejširším a nejjednodušším pojetí, znamená využití informací za účelem tvorby efektivnějších rozhodnutí. Nejčastěji se na definici pojmu BI podílejí společnosti, jež vyvíjejí komplexní BI nástroje, schopné pokrýt veškeré potřebné činnosti od návrhu procesů ETL, přes statistické analýzy až po konečný reporting. Mezi takové společnosti se řadí především Oracle, IBM, Informatica a další.

Definici pojmu BI není třeba dále rozvádět. Komplexní pohled na tento obor nastíní poněkud zjednodušený popis infrastruktury celku.

2.1 Infrastruktura business intelligence

„Infrastrukturu pro aplikace BI tvoří datové sklady, jejichž technologie je v dnešní době významným rysem moderních informačních systémů.“

(Tvrdíková, 2005, s. 104)

Datové sklady jsou v případě BI tedy základním stavebním kamenem. Data, nashromážděna podnikovými informačními systémy jsou ukládána právě do datových skladů. Podrobněji se datovým skladům a přístupům k jejich budování věnuje následující podkapitola.

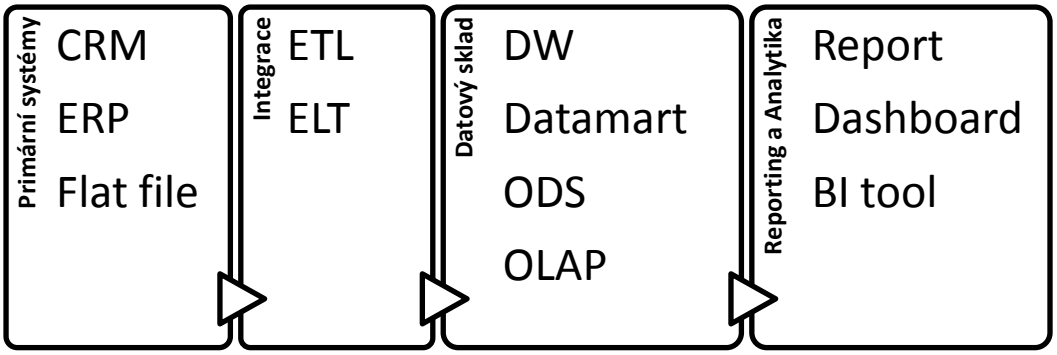
Než se data do datového skladu nahrají, je potřeba, aby byla náležitě transformována do podoby, která podniku bude sloužit nejvhodněji k potřebám analýz nad nimi. Transformace dat je jednou z úloh procesu ETL. Procesu ETL se věnuje podkapitola 2.3.

Tedy důvodem, proč je třeba datových skladů je nejen archivace podnikových dat, ale také analýza nad těmito daty a výsledné výstupní zpracování sloužící k podpoře rozhodování, čili reporting.

Infrastruktura BI jako celku odpovídá krokovému rozložení postupné změny zdrojových dat ve výslednou informaci požadovanou zákazníkem.

Koncepci infrastruktury BI tak, jak je vykreslena na obr. 1, je možné rozdělit do samostatných skupin:

- **primární systémy**, mezi které se řadí např.: CRM[1], ERP[2] a další,
- **integrace**, jež je zastávána procesy ETL či ELT[3],
- **datový sklad**, řadí se zde i datová tržiště (Datamart), ODS[4] či OLAP[5] kostky,
- **reporting a analytika**.



Obr. 1 - Infrastruktura BI, zdroj: autor

[1] CRM - Customer Relationship Management
[2] ERP - Enteprise Resources Planning
[3] ELT – Extract, Load, Transform
[4] ODS – Operational Data Store
[5] OLAP – On-Line Analytical Processing

Jedná se tedy o využívání dat z primárních systémů či zdrojových souborů, která jsou procesem ETL transformována a nahrávána do datových skladů. Nad datovými sklady jsou pak vytvářeny reporty a dashboardy, nebo se data hlouběji zkoumají statickými metodami, čili dochází k dolování dat. K těmto účelům se využívá množství nástrojů. Nástroj, či tool, může podporovat veškeré potřebné funkce od modelování procesu ETL přes analytiku až po reporting, nebo může být pouze úzce profilovaný pouze na jednu z těchto činností.

Následující podkapitoly se věnují výkladu jednotlivých komponent infrastruktury BI tak, jak jsou vyobrazeny na obr. 1. Vzhledem k obsahu této bakalářské práce budou kroky integrace a datových skladů vysvětleny podrobněji později v podkapitolách 2.2 a 2.3.

Pro srovnání je uvedena obecná koncepce infrastruktury BI z obr. 2. Tato koncepce rozděluje BI do několika vrstev. Jsou jimi:

- vrstva pro extrakci, čištění a nahrávání dat,
- vrstva pro ukládání dat,
- vrstva pro analýzu dat,
- prezentační vrstva,
- a vrstva oborové znalosti.

Poslední vrstva je společná celému BI, tedy všem ostatním vrstvám. Obsahem jsou veškeré znalosti, best-practices a know-how ve společnosti.

2.1.1 Primární systémy

Primárními, či zdrojovými systémy (7), jsou myšleny převážně firemní databázové systémy. Pod pojmem systém je pak míněn systém informační, jež je možno označit i jako systém výpočetní (8).

„Informační systém lze definovat jako soubor lidí, metod a technických prostředků zajišťujících sběr, přenos, uchováni, zpracováni a prezentaci dat s cílem tvorby a poskytováni informaci dle potřeb příjemců informaci činných v systémech řízení.“

(Tvrdíková, 2008, s. 18)

Původ samotného slova systém pak vychází z řečtiny spojením slov „syn“ přeloženo jako *dohromady* a „histemi“ jako *seskupovat*. Gála, Pour a Šedivá (2009, s. 23) jej definují takto:

„Systém je účelově definovaná neprázdná množina prvků a množina vazeb mezi nimi, přičemž vlastnosti prvků a vazeb mezi nimi určují vlastnosti chování celku.“

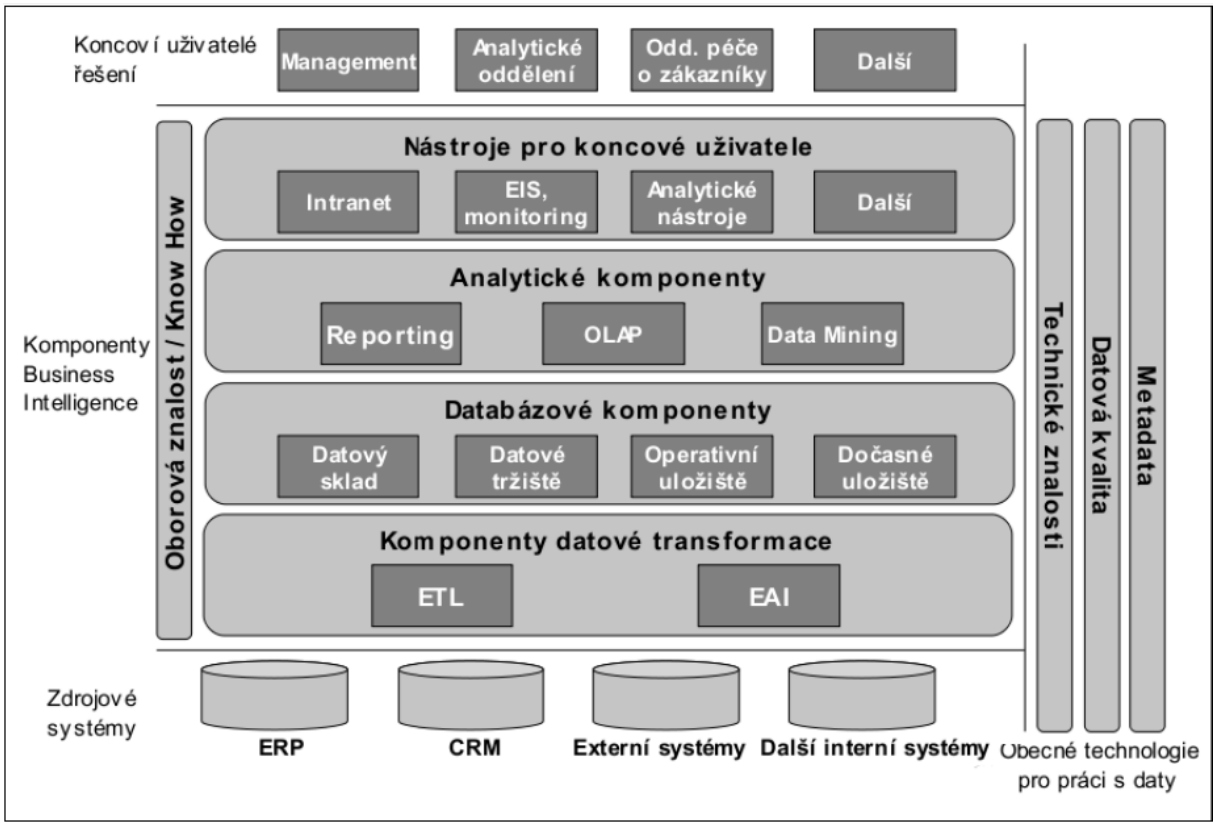
Primární systémy nejsou navrženy pro analytické úlohy, jsou tedy doslova pouhým zdrojem dat. Zdrojová data z primárních systémů jsou procesem ETL nahrávána do datového skladu, kde pak data mohou být analyzována. Příkladem primárních systémů mohou být ERP a CRM systémy, různé úzce profilované systémy sloužící přímo účelům personálního, mzdového či jiného oddělení. Takováto zdrojová data by se dala souhrnně nazvat interní, podniková, avšak jako primární systémy bývají využívány i data nepodniková, tedy ze systémů externích. Mezi takové se řadí telefonní seznamy, výsledky statického úřadu, výstupy státních institucí aj. Externí data jsou často poskytnuty v souborových formátech CSV[6], XML[7] aj. souhrnně nazývaných jako Flat file.

Systémy ERP a CRM spadají do skupiny tzv. OLTP[8] nástrojů. OLTP je transakční databázovou technologií. Takové databáze obsahují velké objemy firemních dat, ovšem svým účelem nejsou primárně určeny k jejich analytickému zpracování. K těmto účelům slouží OLAP. Analytické systémy umožňující rychle zpracovávat a analyzovat data. Zároveň jsou využity ke sledování klíčových výkonnostních ukazatelů tzv. KPI[9], jež slouží pro podporu kvality rozhodování. K analýze OLAP patří tzv. OLAP kostka. OLAP kostku lze popsat jakožto multi-dimenzionálně uspořádaná data, uživateli zobrazována ve dvourozměrné tabulkové formě, s podporou funkcí tzv. dril-down a dril-up, jež slouží pro přechod „*do hlubších*“ úrovní jednotlivých dimenzí. Příkladem může být dimenze času, kdy se z původně zobrazeného časového údaje za rok může uživatel s použitím dril-down dostat na úroveň jednotlivých kvartálů, měsíců, dnů a třeba i podrobněji až po jednotky sekund. Vše záleží na granularitě, tedy zrnitosti dat.

V praktické části této bakalářské práci se s OLAP nepracuje, proto není třeba dalšího rozvádění.

Jako méně obvyklý příklad primárního zdroje dat je možno uvést tzv. message queue. Tyto fronty zpráv, jak by bylo možno daný výraz přeložit, jsou otevřeným komunikačním kanálem, jímž data proudí teoreticky nepřetržitě. V praxi však přenos dat probíhá v dávkách, avšak není dopředu známo, kdy bude nějaký přenos realizován. Proto je třeba, aby byl ETL schopen data “*odchytávat*“ nepřetržitě.

[6] CSV– Comma-sepparated value
[7] XML – Extensible Markup Language
[8] OLTP – On-Line Transactional Processing
[9] KPI – Key Performance Indicator



Obr. 2 - Koncepce infrastruktury BI (9)

Dle koncepce infrastruktury BI, jak ji popisují Novotný, Pour a Slánský (2005), dělící se do několika vrstev, viz obr. 2, jsou primární systémy zařazeny ve *vrstvě pro extrakci, čištění a nahrávání dat*. Tato vrstva zahrnuje i následnou přípravu dat pro jejich následné nahrání do datového skladu, tedy proces ETL.

2.1.1.1 *ERP*

ERP systémy, tedy informační systémy pro plánování podnikových zdrojů, využívány k podpoře automatizace a řízení podnikových procesů a transakcí, podnikových zdrojů a aktivit. ERP systémy byly vyvinuty proto, aby řešily jeden z problémů, se kterým se datové sklady dnes potýkají, a to integraci heterogenních dat. V ERP jsou navrženy tak, aby poskytovaly integrované podnikové řešení, v principu se podobající integrovanému datovému skladu (podkapitola 2.2), umožňující sebemenší entitě podniku, jako účetní oddělení, prodeje, personální oddělení aj. být součástí jedné platformy, databáze i jedné aplikace (9).

2.1.1.2 *CRM*

CRM systémy slouží jakožto informační systémy pro podporu vztahů se zákazníky. V rámci procesu navazování vztahu mezi zákazníkem a společností, hrají důležitou roli informace nashromážděné o zákaznících a schopnost poznat a předvídat jejich potřeby.

CRM demands a contemporary, consistent, and complete image of the customer available to all operational systems that directly or indirectly serve the customer.

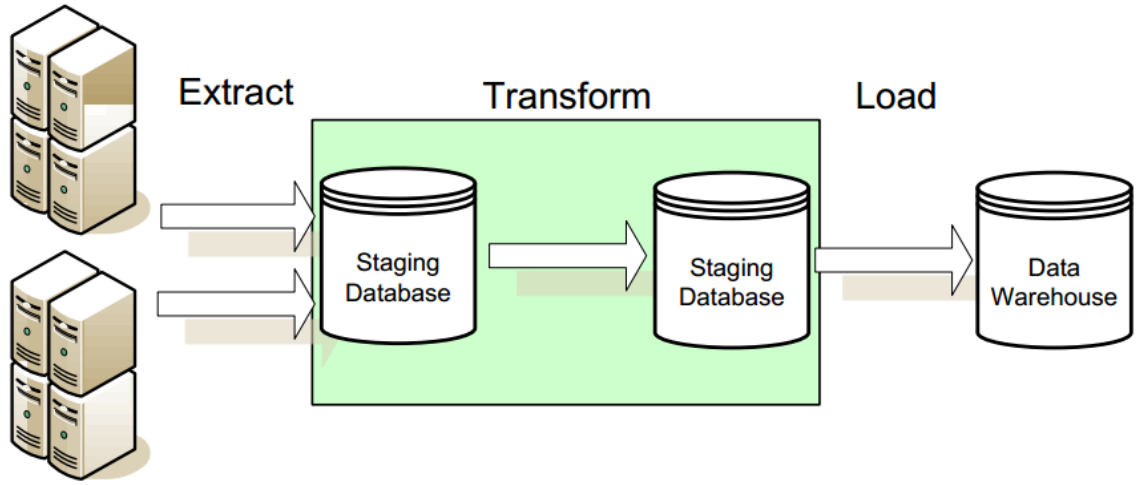
(Kimball, 2004, s. 423)

V systému jsou obsaženy veškeré aktuální údaje o zákazníkovi, čímž poskytují jeho kompletní obraz v rámci dané společnosti. CRM mají tedy za úkol podporovat spolupráci mezi společností a jejími zákazníky k zajištění co nejvyšší oboustranné spokojenosti.

CRM systémy se z pohledu marketingu zaměřují na tzv. 4C, tedy marketingový mix[10]. Cílí především na zákazníka s ohledem na náklady, hodnotu, komfort a komunikaci.

2.1.2 Integrace a Datový sklad

S roztříštěností dat se návrháři integračních procesů setkávají především u firem či společností, provozujících několik oddělení. Jako příklad je možno použít síť maloobchodů využívající rozdílné systémy pro účetní oddělení, sklad a pokladny na každé prodejně. Tato data je v průběhu integračního procesu třeba sjednotit a v požadovaném jednotném formátu nahrát do datového skladu. Datovým skladům se věnuje následující podkapitola 2.2.



Obr. 3 – ETL proces, zdroj: (10)

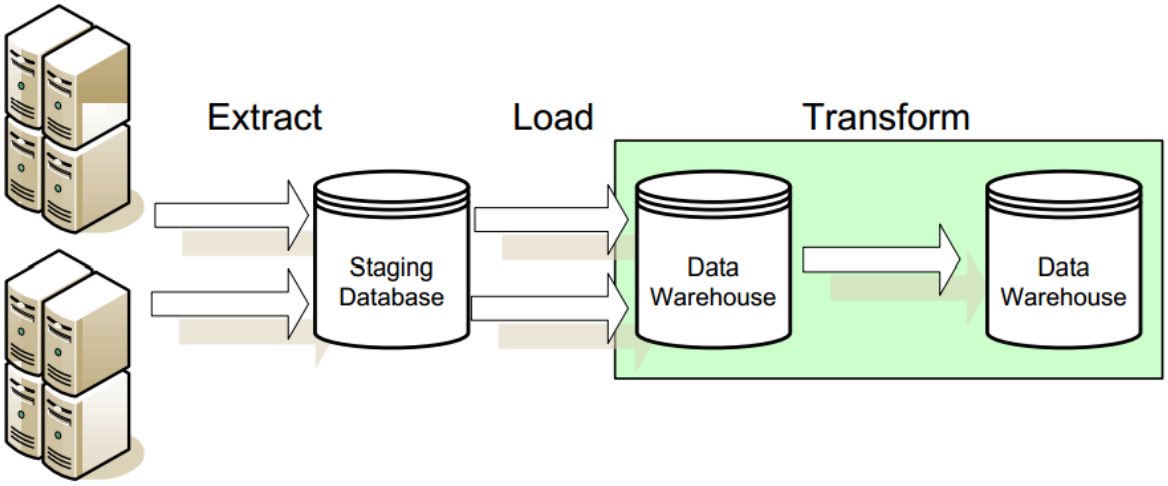
Krok integrace dle znázornění z obr. 1 zahrnuje technologii ETL, jíž se věnuje podrobněji podkapitola 2.3 a technologii ELT, jež je obdobou ETL, avšak pořadí jednotlivých operací procesu se liší.

Jednotlivé kroky ETL, vyobrazeny na obr. 3 jasně definují cestu dat extrahovaných z primárních systémů do dočasného úložiště dat až po jejich nahrání do datového skladu. Celý

[10] Marketingový mix 4C: customer total cost, customer value, convenience a communication; oproti 4P, jež je brán z pohledu podniku, je 4C mixem z pohledu zákazníka.

proces je často navrhován odzadu, tedy požadovaný výstup jako první. Tímto způsobem je jasně řečeno, jaká data jsou požadovány z primárních systémů (9). Dočasným úložištím dat se věnuje podkapitola 2.4.

Přístup k implementaci ELT je na první pohled podobný jako u ETL, obr. 4. Zásadní rozdíl je však v oddělenosti jednotlivých procesů, či kroků. Procesy extrakce a nahrání dat zde nejsou závislé na procesu transformace. Tato oddělenost umožňuje rozdělení projektu do menších bloků, jež jsou lépe zpracovatelné a jejich vývoj a chování lépe předvídatelné. Nevýhodou ELT je stále slabá podpora BI nástrojů (10).



Obr. 4 – ELT proces, zdroj: (10)

Dle obr. 2 s. 13 spadají datové sklady do *Vrstvy pro ukládání dat.* Do této vrstvy jsou zahrnuty datové sklady, datová tržiště, operativní datová úložiště i dočasná datová úložiště (9). Data připravena v datovém skladu jsou dále využívána pro potřeby reportingu a analytiky.

2.1.3 Reporting a analytika

Dle obecné koncepce infrastruktury z obr. 2 s. 13 jsou reporting a analytika rozděleny do vrstev dvou. *Vrstvu pro analýzu dat* a *Prezentační vrstvy*. Není však pravidlem, aby byla data nejdříve použita k analytickým účelům a teprve poté prezentována. Ne zřídka se reporty tvoří rovnou nad datovým skladem, což je jistě poněkud jednodušší, avšak k základnímu přehledu, či sledování KPI mnohdy dostačující. Obr. 1 s. 10 tedy zobrazuje reporting a analytiku z pohledu zpracování dat na stejné úrovni. Data jsou analyzována statistickými metodami, souhrnně nazývanými jako dolování dat, čili data mining.

Koncovému uživateli je však přístupná pouze prezentační vrstva, dle Kimballa spadající do tzv. front room (11). Front room, je výstupní vrstvou v infrastruktuře BI, vrstvou jíž využívají především firemní uživatelé koncového zákazníka. Pozadí, do kterého již tito

uživatelé nezasahují, kde jsou datové sklady budovány a nalévána do nich data pomocí procesů ETL, Kimball označuje jako back room. Tématice back room a front room se podrobněji věnuje podkapitola 2.3.

2.2 Datové sklady

Datový sklad (data warehouse; DW), je úložištěm pro potřeby reportingu, analýz, archivace a konsolidace firemních dat.

Spíše než ke klasickým skladům můžeme datové sklady přirovnat k depozitářům muzeí. I v tomto případě se snažíme shromažďovat exponáty, třídit je časově, geograficky, podle druhu a podobně.

(Lacko, 2003, s. 48)

Z přirovnání datových skladů k depozitářům muzeí je možno odvozovat charakteristiky, jež jsou těmto dvěma porovnávaným subjektům společné. Datový sklad (DS) je úložištěm dlouhodobým, tedy data, nahraná v DS se nelikvidují a slouží k účelům archivace a historizace. Data z primárních systémů jsou do DS nahrávána dávkově a je připouštěna i určitá redundance (12).

Nahraná data musí být dostupná pro dotazování uživatelů. V případech stovek miliónů, či miliardy a více záznamu jsou kladeny vysoké nároky na výpočetní výkon, zastáván desítkami procesorů a kapacitu, mnohdy stovky terabyte. V prostředích DS, typicky u větších společností, probíhá výrazný nárůst dat. Výsledkem je, že k uspokojení rostoucích podnikových požadavků mohou být potřeba další kapacity. DS by měl být tedy navržen tak, aby splňoval tyto požadavky na flexibilitu. Návrh a realizace datového skladu je dlouhodobým procesem, v této bakalářské práci souhrnně nazývaným jako budování DS, kdy délka trvání dosahuje i několika let.

Při návrhu DS není na místě, aby byl koncový uživatel, či zákazník, návrhářem tázán otázkou: „Co *chcete mít ve svém datovém skladu?“* Tímto by byl zákazník stavěn do role samého návrháře DS, zároveň je mu takto dána zodpovědnost za něco, k čemu nebude s největší pravděpodobností způsobilý, tedy by bylo samotnému procesu návrhu přistupováno poněkud neprofesionálně. Po vyspecifikování požadavků zákazníka na DS je na návrhářích aby určili, jakým způsobem se bude při budování postupovat. Jak již bylo zmíněno v úvodu této kapitoly, existují dva druhy přístupů k budování DS. Přístup integrovaného a přístup dimenzionálního modelu.

2.2.1 Integrovaný model

„Datový sklad je podnikově strukturovaný depozitář subjektově orientovaných, integrovaných, časově proměnlivých, historických dat, použitých na získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data."

(Lacko, 2003, s. 48)

Bill Inmon přišel v 70. letech s pojmem DS a svým přístupem k jejich budování dal za vznik myšlence integrovaných DS (EDW[11]). Definice dle knihy *Data Warehouse Toolkit* (Inmon, 2005) popisuje DS jako „*integrovaný, subjektově orientovaný, stálý a časově rozlišitelný souhrn dat, uspořádaný pro podporu potřeb managementu.*" Rolí EDW je příprava a integrace dat tak, aby dávaly komplexnější obraz analyzovaného subjektu. V rámci EDW se tedy odehrává onen krok, který „*mění data v informaci"* (12).

DS je jednou z částí celého systému BI. Podnik má jeden jediný DS, který je dále zdrojem informací pro datová tržiště (13).

EDW lze popsat jako shromáždiště veškerých korporátních dat, ze všech regionů kde daná společnost figuruje, firemních poboček či oddělení. Jedná se o veškerá nashromážděná data na jedné hromadě, kdy je tímto utvářen celkový datový obraz podniku. Struktury dat jsou uspořádány v hierarchii tak, aby odpovídaly časově a archivačně potřebám EDW. EDW je primárně využíván k dotazování nad daty a jejich analýzu při současném nezatěžování primárních systémů.

Inmonův přístup řeší archivaci dat a jakým způsobem uchovat historii. Archivací se rozumí ponechání starého záznamu ve strukturách DS, tedy záznam se nelikviduje. Historizace je spojena s platností daného záznamu a často je řešena pomocí časových značek. Inmon se svým přístupem budování DS zároveň kloní na stranu, kdy je vedle zdrojových systémů navržena nová databáze, již zoptimalizovaná pro výkon tak, aby rychlost odezvy při dotazování se nad uloženými daty byla co nejvyšší. Rychlost odezvy dotazu je samozřejmě relativní a závisí na počtu propojených tabulek potřebných k získání dotazovaného výsledku. Tato optimalizace stojí především na normalizaci, či případně de-normalizaci, dat. Není potřeba uchovávat data ve vyšších normálních formách (NF) než ve 3NF, tedy není potřeba žádného číselníku ve zvláštních tabulkách kvůli zajištění referenční integrity. Místo toho se hodnoty, které by bylo třeba přebírat pomocí cizích klíčů, rovnou uloží v normalizovaném prostoru 3NF.

[11] EDW – Enterprise Data Warehouse

Normalizace dat je procesem používaným k odstranění anomálií v relačním datovém modelu. Její podstatou je postupná dekompozice datového modelu rozdělením atributů do většího počtu relací. Zpravidla platí, že čím jsou tabulky relační databáze ve vyšší formě, tím snáze se s nimi pracuje (7).

K základním vlastnostem relace patří neexistence duplikátů n-tic; libovolné pořadí atributů; libovolné pořadí n-tic; nerozložitelnost hodnot atributů (14). Tedy žádný atribut nelze dále rozložit, aniž by došlo ke ztrátě informace, jedná se o atomičnost atributu (15). Normalizace dat je procesem postupným a dělí se dle úrovně normalizace do jednotlivých forem.

V 1NF se relace nachází, pokud jsou dodrženy principy její definice.

Ve 2NF je relace, když je v 1NF a každý neklíčový atribut je plně funkčně závislý na primárním klíči relace.

Ve 3NF je relace, jestliže je v 2NF a každý neklíčový atribut je netranzitivně závislý na primárním klíči. (15)

Data je možné dále normalizovat až do 5NF, Inmon však ve svém přístupu využívá nejvýše 3NF.

Mezi nevýhody Inmonova přístupu patří vysoká časová náročnost celého projektu budování DS a vysoké náklady na projekt. Vzhledem k robustnímu charakteru EDW, kdy je DS v Inmonově pojetí i možné charakterizovat jako rigidní, je při plánování pořizování EDW třeba počítat s dobou dodávky až několika let. V době dodávky by tedy EDW neodpovídal zcela skutečnosti, proto je potřeba neustálé aktualizace.

Důvody pro takto dlouhé trvání projektu jsou prosté. V první řadě je to lidský faktor. Jakým stylem je nastavena daná společnost či organizace, jaké jsou definující role jednotlivých členů BI týmu a schopnosti vše řádně dokumentovat. V projektovém řízení vždy existuje proces definující projektové zadání a business specifikace, nad kterým je pak provedeno technické řešení (návrh tabulek, návrh vazeb, návrh schématu), které je pak třeba vyvinout. Nad navrženými tabulkami se poté navrhuje a implementuje ETL.

Ať už z pohledu projekt managementu, kdy bývá často k vývoji přistupováno vodopádovou metodou (waterwall method) namísto vhodnějšího agilního vývoje, přes neochotu uživatelů po špatné alokování zdrojů. Zmíněná vodopádová metoda vývoje zdržuje budování DS především v neustále se opakující validaci každého kroku a zároveň i při samotném návrhu, který stojí především na specifikacích požadavků zákazníka, jež se však

mohou postupem času lišit. V takovém případě může trvání budování DS trvat i 3-4 roky. V době dokončení již výsledek nemusí zcela splňovat představy zákazníka a nebude odpovídat realitě. Náklady takto dlouhotrvajícího projektu náročného na zdroje se pohybují v řádu desítek až stovek miliónů korun.

V praxi pak postup vypadá následovně. Je navržena první verze datového modelu. K návrhu se využívá modelovací BI tool (např.: Power Designer). V hotovém modelu jsou DS (a účelům, kterým bude sloužit) na míru navrženy tabulky, ve kterých budou uchována strukturovaná data z primárních systémů. Tyto tabulky jsou dále vyexportovány do testovací databáze, kde jsou nad nimi následně připravovány předpisy pro mapování ze systémů primárních. Dle těchto předpisů jsou následně realizovány procesy ETL.

Čím komplikovanější tool, čím větší tým a jeho geografické rozložení, tím je konsolidace dat, testování a integrace celku nákladnější a zdlouhavější.

Zmíněné aktualizace EDW, jež jsou potřeba pro zajištění, aby EDW v době dokončení odpovídal realitě a požadavkům zákazníka, jsou druhým důvodem jeho dlouhotrvajícího vývoje.

Primární systémy procházejí aktualizacemi a jsou vyvíjeny jejich nové verze (release). V případě budování EDW je během postupného vývoje třeba tyto verze implementovat a udržovat tak EDW aktuální. Nové aktualizace mají často za následek různé změny v tabulkách, nad kterými jsou již navrženy ETL procesy. V případě jakékoliv změny je třeba nového návrhu ETL procesu, kdy budou při návrhu brány aktualizace v potaz. Zároveň takto existuje paralelně několik verzí EDW.

Existují snahy agilního vývoje EDW. V případě agilního vývoje je definována pouze součást celku, jež je schopna fungovat samostatně, tedy modul, a nechávají se „zadní vrátka“, kudy se budou moci připojit další moduly DS, jež budou vyvinuty až poté, např.: nejdříve je vyvinut modul pro oblast financí. Aby již bylo možno poskytovat výsledky a s nimi pak dále nakládat ať už pro účely prezentace či analýzy, je třeba, aby se s tímto modulem mohlo samostatně pracovat. Poté, co jsou vyvinuty moduly další (produkční, zákaznický) a jsou přidány další tabulky, se těmito vrátky moduly propojí a DS se rozrůstá. V praxi to znamená kladení vysokých nároků na lidské zdroje v ohledu kooperace vývojových týmů. Je třeba tým, který udržuje v produkci onen finanční modul. Zároveň je mezitím vyvíjen modul zákaznický. Než jsou tyto moduly napojeny je třeba vyvíjený modul řádně otestovat. Pro ten je třeba udržovat testovací prostředí, které odpovídá prostředí produkčnímu. Vše se komplikuje v případě, kdy je aktualizován již modul, který je v produkci. V takovém případě je pomocí patchů třeba ošetřit a aktualizovat moduly, jež byly teprve vyvíjeny či testovány pro napojení.

Z příkladu lze vyčíst obtíže vznikající při vývoji EDW, jež zapříčiňují jeho dlouhodobý vývoj. Z pohledu zákazníka je délka trvání jistě negativní, naopak z pohledu firmy DS vyvíjející jistě pozitivní.

Právě na tyto problémy se svým přístupem budování DS zaměřil Ralph Kimball. Kimball prosazuje dimenzionální model (6), jež stojí na datových tržištích oproti centrálnímu datovému úložišti Inmonova pojetí.

Integrovaný datový sklad je centrální datové úložiště, kde je požadavek konzistence naprosto zásadní (DS musí poskytovat „jedinou verzi pravdy“)

(Tvrdíková, 2008, s. 108).

Nutnost zajištění konzistence dat, čili jejich sourodosti, je ve výsledku výhodou Inmonova přístupu vzhledem k přístupu konkurenčnímu. V Kimballově přístupu, kdy je prosazováno budování samostatných datových tržišť, jakožto nadstavby nad základním EDW, jsou právě možná nekonzistence dat a komplikované načítací procesy nevýhodou.

2.2.2 Dimenzionální model

Je důležité říci, že dimenzionální přístup není vynálezem jen jedné osoby. Pojmy fakt a dimenze pocházejí již ze 60. let, kdy byly použity ve společném výzkumu spolčenosti General Mill a Darthmouthské univerzity (6).

Ralph Kimball přišel v 90. letech s dimenzionálním přístupem budování DS. Přístup se od integrovaného modelu Billa Inmona liší především kratšími časovými nároky na návrh a implementaci DS a ve srovnání s rigidností integrovaného modelu působí značně odlehčen. Pokud je Inmonův přístup, vzhledem k užívání nejvýše 3NF již označován za de-normalizovaný, pak je možné přístup Ralpha Kimballa popsat jako silně de-normalizovaný. Často je používáno pouze 2NF. Záleží na typu schématu požitého při datovém modelování. Redundance v datech jsou však kompenzovány relativně rychlou odezvou dotazování. Použití dimenzionálního přístupu je preferováno i netechnicky orientovanými uživateli (16). Dimenzionální přístup budování DS je určen pro optimalizaci databází za účelem podpory rozhodování v rámci rozsáhlých dotazů.

Dimenzionální model využívá rozdělení dat na dimenze a fakta, ty jsou uchovávána v tabulkách dimenzí (referenční tabulky) a faktů. Dimenze je chápána jako rozměr nad daty. Může jí být region, kde provozuje daná společnost své pobočky a může jí být produkt či produktová řada. Tabulka faktů je pomoci cizích klíčů s tabulkami dimenzí propojena a je

schopna při dotazování rychle vracet výsledky odpovídající požadované kombinaci dimenzí, např.: počet prodaných produktů X z produktové řady Y v regionu Z.

V tomto přístupu jsou při návrhu DS brány v potaz business aspekty zákazníkovy společnosti a ohled na styl projektového řízení. DS poskytuje každému oddělení firmy zvlášť výseč potřebných dat. V takovém případě mluvíme o datových tržištích. Ty jsou v dimenzionálním přístupu nadstavbou datového skladu, tedy nadstavbou Inmonova EDW. DS v Kimballově pojetí je konglomerátem všech datových tržišť v rámci podniku.

Avšak postup budování je opačný. Otázka budovaní DS je volbou mezi EDW, tedy shromážděnými daty celé společnosti v jednom centralizovaném skladu, kdy se teprve nad těmito daty dále budují datová tržiště, nebo se nepouštět do budování jednoho velkého a časově náročného EDW a místo toho se zaměřit pouze na vybudování úzce zaměřených datových tržišť, poskytujících pouze potřebnou výseč dat z oblasti zájmu daného oddělení společnosti, ať už se jedná o oddělení účetní a mzdové, logistické či personální.

Prakticky, při budování DS dimenzionálním přístupem, se nejprve utvoří podkladový dimenzionální DS a poté je již věnována pozornost převážně k budování datových tržišť. Datová tržiště jsou tedy agregovanou vrstvou nad podkladovým DS, navržena k zvládnutí vysokého výpočetního výkonu. V případě potřeby implementace aktualizace DS (release) se jedná ve srovnání s aktualizací integrovaného přístupu o relativně malý zásah, jež by se dal popsat jako lehká změna v tabulce faktů. V týmu jsou potřeba pouze 2-3 lidi, kteří se starají o chod při spouštění nového release.

"A data warehouse is a system that extracts, cleans, conforms, and delivers source data into a dimensional data store and then supports and implements querying and analysis for the purpose of decision making."

(Kimball, 2004, s. 23)

Z dané definice vyplývá významnost dimenzionálního datového skladu v souvislosti s podporou dotazování a analýz k účelu zlepšení kvality rozhodování. V následujícím textu budou vysvětleny pojmy dimenze a fakta. V následující podkapitole (2.3) je věnována pozornost první části výše zmíněné definice a to procesu ETL, v Kimballově pojetí označován i jako ECCD[12], jež slouží nahrání dat do DS.

[12] ECCD – extract, conform, clean, deliver

2.2.2.1 *Dimenze*

Tabulky dimenzí poskytují obsah tabulkám faktů. Ve srovnání jsou však tabulky dimenzí obsahově menší než tabulky faktů. Dimenze je chápána jako rozměr nad daty, umožňující zjednodušenou organizaci dat pro jejich přehlednost. Avšak platí zde tvrzení, že DS je pouze tak dobrý, jako jeho dimenze (11).

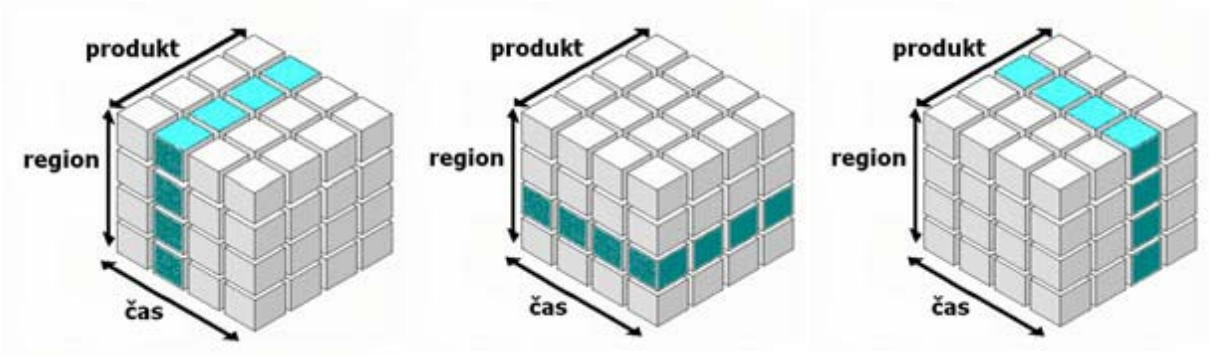
Dimenze je strukturou, často se skládající z jedné nebo více hierarchií, které kategorizují data. Kombinace několika různých dimenzí s fakty umožňuje odpovídat na dotazování ze strany vedení firmy.

Dimenzionální tabulky poskytují strukturované popisné informace k jinak neuspořádaným číselným metrikám. K tomuto popisu je využíváno dimenzionálních atributů. Dimenze mohou být velmi rozsáhlé a obsahovat tak i 100 atributů. Dimenzionální atributy tedy slouží jako primární zdroj při dotazování v rámci datového modelu a jsou pomocí primárních klíčů dané dimenze propojeny s cizími klíči tabulky faktové.

Mezi nejčastěji používané dimenze patří čas, produkt, zákazník a region. Data v dimenzích jsou uchovávány s nejvyšší úrovní podrobností, z pohledu hierarchie tedy data na úrovni nejnižší. Poté jsou agregována, sumarizována. Ve formě těchto úhrnů jsou hierarchicky na úrovni vyšší, avšak méně podrobné. Data na vyšších úrovních jsou užitečná k analýzám. Hierarchie jsou logickými strukturami organizujícími úrovně dat. Příkladem hierarchie může být dimenze času: minuta – hodina – den – týden – měsíc – kvartál – rok.

„Dimension tables are the entry points into the fact table. Robust dimension attributes deliver robust analytic slicing and dicing capabilities.“

(Kimball, 2002, s. 20)



Obr. 5 - Řezy kostkou podle časové, regionální a produktové dimenze, zdroj: (7)

Jednoduše uchopitelným nástrojem dimenzionálního modelu je OLAP kostka. Kostka je však jen obrazné pojmenovaní pro jednoduchou interpretaci a vysvětlení dimenzionálního modelu. Tři dimenze (jako v případě kostky) nejsou pravidlem, OLAP kostka může být

složena z mnohem většího počtu dimenzí (záleží na nástroji, ve kterém je OLAP kostka vytvářena, např.: kostky vytvořené v MS OLAP Services mohou obsahovat až 64 dimenzí). Obr. 5 zobrazuje způsob „porcování“ OLAP kostky z pohledu jednotlivých dimenzí času, regionu a produktu. Nad OLAP kostkou je možné dotazování pomocí jazyka MDX, jež je vícerozměrným dotazovacím jazykem.

2.2.2.2 *Fakta*

Faktová tabulka je jádrem dimenzionálního modelu. Tabulka faktů obvykle obsahuje dva typy sloupců, ty jež uchovávají numerické hodnoty, čili metriky a sloupce, jež uchovávají cizí klíče napojené na dimenzionální tabulky. Řádek faktové tabulky vyjadřuje určitou metriku či hodnotu, jež je logicky svázána s několika dimenzemi přes cizí klíče.

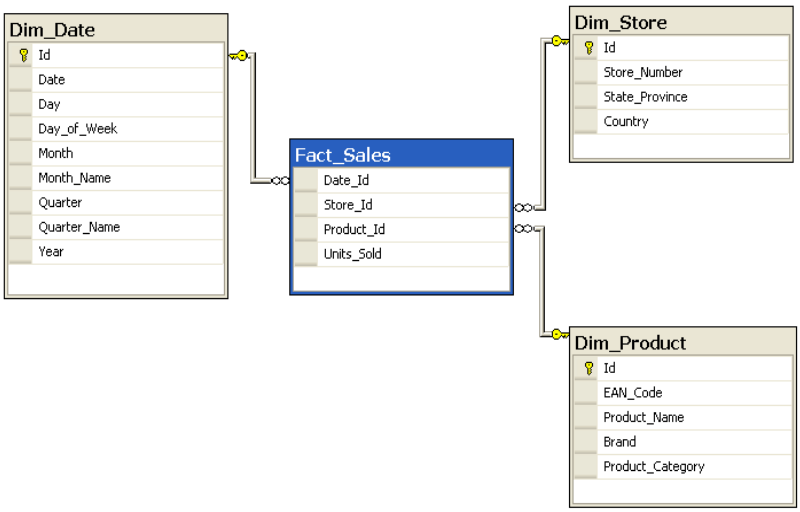
Faktová tabulka je primární tabulkou dimenzionálního modelu. Obsahuje veškeré podnikové výkonnostní ukazatele či metriky. Tyto metriky jsou většinou numerického charakteru, ovšem je možné se setkat i metrikami textovými. Textové metriky jsou často popisné, kdy k tomuto popisu je využito pouze omezeného předem určeného seznamu povolených hodnot. Pojem fakt je používán jako vyjádření podnikové metriky. Veškeré metriky ve faktové tabulce musí mít stejnou zrnitost či granularitu (16). Granularitou rozumíme stupeň podrobnosti uložených dat. Faktové tabulky by měly obsahovat nejjemnější detaily o datech, tedy mít vysoký stupeň granularity. Mezi nejužitečnější fakty se řadí ty, jež jsou zároveň numerické i aditivní. Numerické metriky mohou být údaje o vyplacené částce (např.: v dolarech), množství prodaného produktu či číselné označení oddělení nebo zaměstnance aj. Aditivnost je klíčovou vlastností faktu. Tato vlastnost umožňuje k sobě „přidat“ dvě rozdílné metriky a přesto dostat vypovídající výsledek. Např.: sloučení metriky počtu prodejů s pobočkami vrátí zpět údaj o počtu prodaných kusů na pobočce X. Pokud je k tomuto přidána i časová dimenze, výsledný údaj by vypovídal o počtu prodaných kusů na oddělení X za měsíc M. Metriky mohou být i semi-aditivní, jež je možno přidat pouze k určitým dimenzím a neaditivní.

Faktové tabulky obsahují dva nebo více cizích klíčů, jež odpovídají primárním klíčům v tabulkách dimenzionálních. Zároveň vyjadřují vztahy m:n mezi jednotlivými dimenzemi dimenzionálního modelu. Svou velikostí, tedy počty uložených údajů, faktové tabulky zabírají 90 % procent z celkové velikosti dimenzionálního modelu.

Struktura dimenzionálního DS je určena propojením tabulek dimenzí a faktů. Jsou rozlišovány dva hlavní typy napojení dimenzí na faktovou tabulku, jedná se napojení ve hvězdě (star schema) a vločce (snowflake schema).

2.2.3 Schémata datového skladu

Hvězdicové schéma (*star schema*) je vůbec nejčastěji využívaným způsobem propojení dimenzionálních a faktových tabulek. Jedná se o převedení relačního modelu na model dimenzionální. Centrální tabulkou schématu je tabulka faktová, jež je propojena pomocí cizích klíčů s doplňujícími dimenzionálními tabulkami.



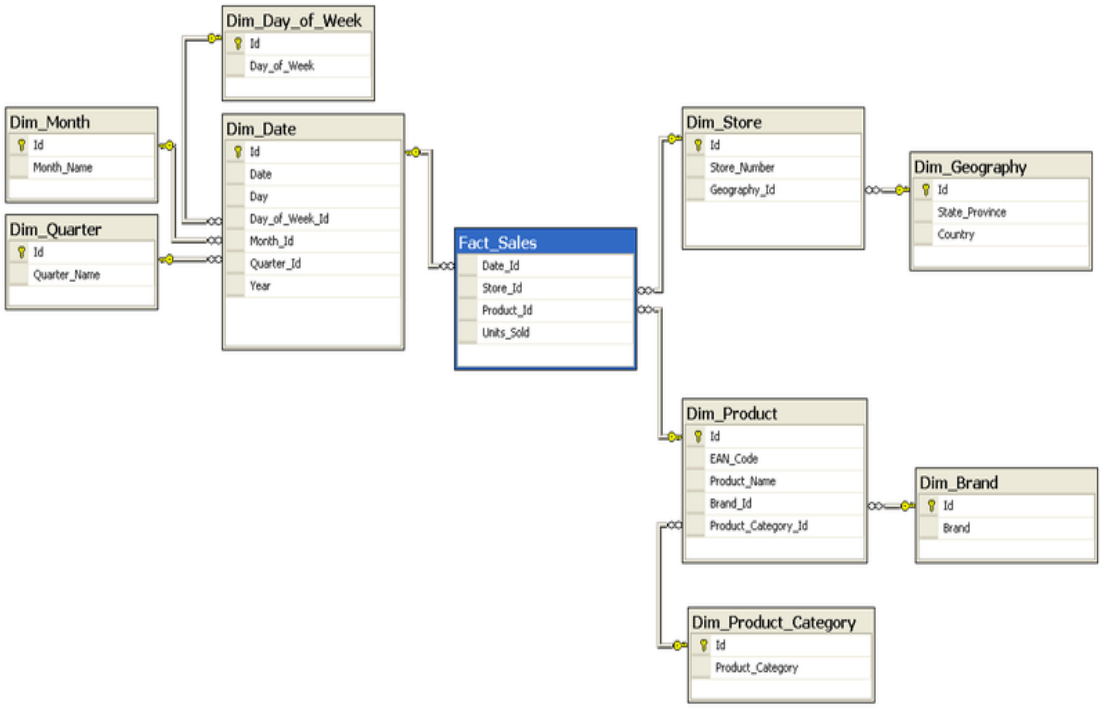
Obr. 6 - Star schema, zdroj: http://en.wikipedia.org/wiki/File:Star-schema-example.png

Název schématu vychází z tvaru, jež uskupení dimenzionálních tabulek okolo tabulky faktové v grafickém vyjádření tvoří, viz obr 6. Každá dimenze je reprezentována jedinou tabulkou typicky obsahující dimenzionální atributy.

Tabulky dimenzí ve star schema jsou často de-normalizovány pro dosažení vyšší rychlosti dotazování ve srovnání se schématem vločky.

Schéma vločky (*snowflake schema*) je možno popsat jakožto normalizované star schema. Normalizovány jsou tabulky dimenzí. Na ty jsou opět pomocí cizích klíčů připojeny číselníky a výsledný rozvětvený tvar při grafickém vyobrazení schématu svou strukturou připomíná sněhovou vločku, viz obr 7. Díky normalizaci je možno procházet dimenzionální tabulky s vysokou granularitou, kdy jsou podrobné detaily získávány z číselníků. Zároveň je snížena redundance dat, tedy i nižší nároky na diskový prostor. Jak bylo zmíněno výše, faktová tabulka dosahuje i 90 % veškeré kapacity tabulek v DS. Ušetřený diskový prostor díky použití snowflake schema je tedy zanedbatelný.

Kimball sám nedoporučuje používat snowflake schema z důvodů re-normalizace dimenzí do 3NF, jež je často, dle Kimballa mylně, ospravedlňováno výsledným zlepšením udržovatelnosti dimenzí, zvýšení flexibility a nižších nároků na diskový prostor (17).



Obr. 7 - Snowflake schema, zdroj: http://en.wikipedia.org/wiki/File:Snowflake-schema-example.png

V praxi je možné se setkat s podrobněji rozděleným typem star schema. V případě, že je využíváno více faktových tabulek, jež sdílejí stejné dimenze, je celkové schéma někdy nazýváno souhvězdím (constellation).

2.3 ETL

ETL, čili proces extrakce, transformace a nahrávání dát, v české odborné literatuře také nazýván datovou pumpou (7) (9), je komplexním procesem, potřebným k přípravě dat pro cílové využití firemními uživateli, kteří pak na připravených datech zakládají strategická rozhodnutí.

„ETL process is the sequence of applications that extract data sets from the various sources, bring them to a data staging area, apply a sequence of processes to prepare the data for migration into the data warehouse, and actually load them.“

(Loshin, 2003, s. 146)

Jedná se tedy o sekvenci kroku, potřebných k extrakci dat z primárních systémů, jejich přesunu do DSA, transformace těchto dat a jejich konečné nahrání do struktur DS (1). Významem ETL je vybudovat z hlediska nákladů efektivní, spolehlivý, rozšiřitelný, kompatibilní, pozorovatelný, zabezpečený, snadno ovladatelný systém pro zavedení údajů do DS a připravovat je tak pro dotazování koncového uživatele (11).

Existují dva způsoby jak proces ETL vykonávat. Buď s pomocí příslušných ETL nástrojů (např.: Informatica, Microsoft SSIS, DataStage a jiné), nebo za pomoci příkazů operačního systému (OS) v kombinaci s dotazovacím jazykem.

Snadná ovladatelnost je především doménou grafického uživatelského rozhraní ETL nástrojů. Tyto nástroje však nemusí podporovat veškeré potřebné funkce a tak je možné narazit na různé limitace. Příkladem může být potřeba manipulace s Flat file, jež zrovna není zcela daným nástrojem podporována. V takovém případě je třeba využívat příkazů OS. Obecně je možno tvrdit, že veškeré operace s daty, jež zastávají ETL nástroje, je návrhář schopen vytvořit pomocí kombinace příkazů OS a dotazovacího jazyka. V takovém případě je třeba odpovědět na otázku, proč je třeba ETL nástrojů?

V zásadě kombinace těchto dvou typů příkazů zabezpečí opravdu vše. Výhoda ETL nástroje je však krom zmiňovaného user friendly grafického uživatelského prostředí (GUI[13]) v jeho výkonnosti a to i přesto, že se problém s určením výkonnosti může jevit těžko porovnatelný. Výkonnost závisí na objemu zpracovávaných dat, používaném hardwaru i na tom, jakým způsobem je databáze s uloženými daty nainstalovaná. Je možné, aby na dva servery, stejně vybaveny hardwarově, s instalací stejné databáze a za použití stejného ETL

[13] GUI – Graphical User Interface

nástroje, prokazovaly rozdílnou výkonnost ve zpracování dat. Přitom jediný rozdíl může být v jednoduchém nastavení databáze a výpočetní výkon vzroste až několikanásobně.

V praxi se používá obou způsobů. V případě, kdy jde o ETL striktně z Flat file do Flat file, pak by mohlo být využíváno pouze příkazů primárního OLTP systému, např.: příkazy operačního systému UNIX (script shell). Tento případ je možno příhodně nazvat jako nízkonákladové ETL, vzhledem k nepotřebnosti nákladných nástrojů. Druhý uvedený způsob tedy teoreticky nepotřebuje jazyk dotazovací, avšak v praxi se transformace nad daty neprovádí přímo v primárních systémech. Data jsou nejprve extrahována do dočasného úložiště dat (podkapitola 2.4) a dále zpracovávána až tam.

Ať už se jedná o ETL s použitím nástrojů, nebo ne, v obou případech je zahrnut sběr dat s rozličných zdrojů, jejich validace k zajištění přesnosti dat, následné čištění dat a zajištění jejich konzistence, či případné přizpůsobení dat určitým business rules dané společnosti. Čištěním dat se rozumí odstranění překlepů, převedení na shodné formáty, případné doplnění chybějících hodnot. Jejich konečné nahrání do DS či datového tržiště pak slouží pro další analýzy, reporting a dotazování dat ze strany firemních uživatelů.

Ke správnému porozumění zkratky ETL je třeba chápat jednotlivé části:

- E jako **Extract** – Extrakce dat do back room DS,
- T jako **Transform** – Transformace dat,
- L jako **Load** – Nahrání těchto dat do prezentační vrstvy DS.

Každá z těchto tří operací je na následujících stránkách věnována samostatná podkapitola.

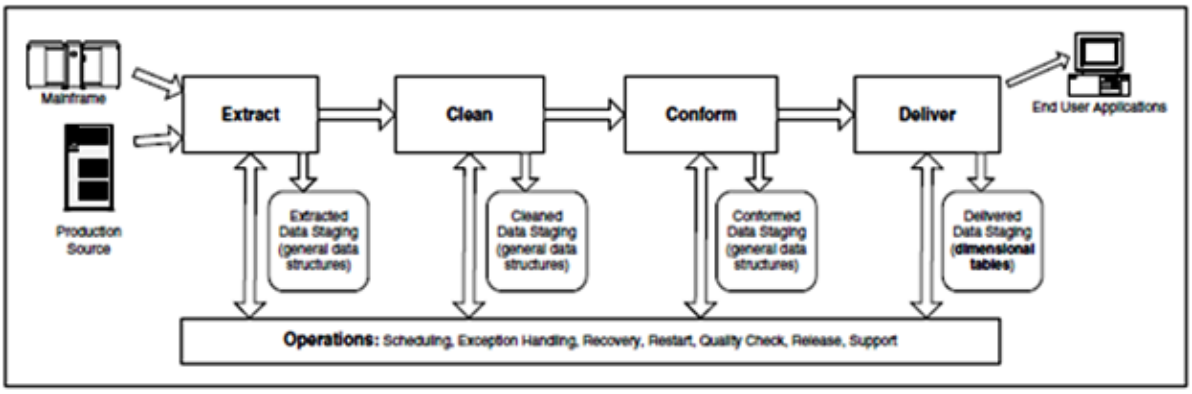
K vysvětlení účelu ETL lze použít přirovnání celého procesu k použití síta na stavbě. Celá analogie je pouze obrazná a čtenářem by neměla být chápána jako přesné vysvětlení funkčnosti jednotlivých kroků ETL.

Tak jako je požadavkem stavbyvedoucího, je i v zájmu zákazníka, aby materiály pro stavbu domu byly co nejvyšší kvality. Štěrk, jež bude na stavbě využíván, je vhodné před použitím protřídit přes síto za účelem odstranění nekvalitních zrn a nečistot. To, jak je zde definován pojem kvality, záleží na typu síta, které je používáno. Důležité je odfiltrovat kamení, listí, trávy či nečistoty, které mohou kvalitu štěrku výrazně snižovat. Je však chybou, pokud jsou kladeny vysoké nároky na kvalitu zrn a samotné síto tyto požadavky nesplňuje. Výsledná protříděná zrna by tak neodpovídala požadavkům a celá kvalita stavby, která se od tohoto dovíjí, by byla tímto ovlivněna. Proto je třeba, aby síto bylo navrženo a sestrojeno kvalitně.

Štěrk lze analogicky srovnat s daty v primárních systémech. Taková data jsou často nesourodá, obsahují redundance či duplicity, neshodují se ve formátu a jsou uložena na rozdílných platformách.

Business intelligence specialistu, který má na starosti procesy ETL, je tedy možno si představit jako strůjce síta.

Kimball (2004) podrobněji definuje ETL jakožto proces o krocích čtyřech a nazývá jej ECCD[14]. Jak ETL tak ECCD procesy, jsou shodné a pouze u druhého z nich je přiznávána vyšší důležitost samotnému kroku transformace. Transformace je tedy rozdělena na kroky dva, vyčištění dat a jejich přizpůsobení, viz obr. 8.



Obr. 8 - ECCD, zdroj: (11)

2.3.1 Extrakce

Prvním krokem datové integrace je úspěšná extrakce dat z primárních systémů. Vzhledem k tomu, že ve společnosti bývají používány rozdílné operační systémy, databázové systémy, běžící na jiném hardwarovém vybavení, ba dokonce samotné pobočky či oddělení jedné a té samé společnosti jsou často na sobě nezávisle v ohledu použití operačního i databázového systému, je třeba při návrhu procesu extrakce brát tato omezení v potaz. Kimball (2004) hovoří o logické a fyzické nekompatibilitě, kterou má ETL za úkol řešit efektivní integrací systémů s rozdílným:

- databázovým systémem,
- operačním systémem,
- hardwarem,
- komunikačními protokoly.

[14] ECCD – Extract, clean, conform, deliver

Extrahují se tedy data z databáze nebo Flat file, případně z message queue. Data jsou extrahována v dávkách a to i přes jejich teoreticky nepřetržitý tok. Toto rozporcování dat na jednotlivé dávky (tzv. batch) probíhá z důvodu nemožnosti ETL zpracovávat data nepřetržitě. Pro potřeby ETL je periodicky prováděno ukončení dávky a každá dávka je zpracovávána samostatně.

Data jsou extrahována do dočasného úložiště dat (podrobněji v podkapitole 2.4). Dočasné úložiště dat je nedílnou součástí ETL. Návrháři ETL nemusí mít vždy garantovaný přístup k datům v primárních systémech a i v případě, kdy by tento přístup garantován byl, je nemyslitelné, aby se prováděly nad daty transformace za chodu těchto systémů. Případné problémy či chyby zaviněné špatně navrženou transformací, by tak mohly zavinit poškození dat primárních systémů. Z tohoto důvodu jsou veškerá zpracovávaná data nejprve extrahována a nahrána v poměru 1:1 do dočasného úložiště dat.

2.3.2 Transformace

Transformace je krokem úpravy dat, jež (s použitím notné generalizace) řeší především dvojí potřebu. Manipulaci s daty a obecně interní pravidla BI.

K manipulaci s daty patří, jednak schopnost data kombinovat z různých zdrojů, možnost nad daty dělat agregace, aplikovat filtry, nebo data typicky transformovat podle vzorců. V případě nadefinovaných vzorců jsou předem známa všechna data, s nimiž bude ve vzorci počítáno, za účelem potřeby znát výsledek. Příkladem může být výpočet úroku vkladu na bankovním účtu.

V ideálním případě je možno předdefinovat chyby, které mohou během transformace nastat a k nimž se nadefinují operace, které vedou k jejich nápravě. Typicky to mohou být různě zkomolená jména.

Ideální případ by obsahoval databázi zkomolenin, dle které by se přicházející záznamy opravovaly. Ovšem případů, kdy je návrhář schopen předdefinovat tyto chyby, není mnoho a naprostá většina případů je taková, kdy je zjištěno, že nějaký údaj chybí úplně, či nelze dohledat vazbu. V takovém případě nelze určit, zdali záznam ještě uložen není, nebo zdali uložen je, avšak nějak zkomolený.

Ovšem ideál je daleko od praxe a v dnešní době probíhá trend přesunu kroku čištění dat, jež je dle obr. 8 s. 28 krokem druhým, spíše do specializovaných nástrojů, tak jako data profilig.

Mezi interní BI pravidla, či pravidla DS, patří *primární klíče dimenzí*, tzv. surrogate key. Surrogate key (SK) nahrazují (překládají) primární klíče a jsou tedy jakýmsi zástupným

umělým klíčem pro potřeby propojení faktů s dimenzemi. Primární klíč (PK) je využíván k jednoznačné identifikaci záznamu v databázi. Příkladem je číslo bankovního účtu. Takové číslo účtu musí být vždy jedinečné, nejen v rámci dané banky, ale i mezi bankami.

S překladem PK je dosahováno navýšení dotazovacího výkonu. K tomuto dochází především z důvodu vhodnějšího užití datového typu u SK, jež jsou typicky integer, či obecně číselným klíčem. PK v praxi často číselného datového typu nebývá (nejčastěji je možno se setkat s datový typem varchar / nvarchar), ovšem pravidlo překladu je aplikováno i pokud PK integer je. Ke každému PK se tedy generuje SK a to i v případě, kdy samotný PK je číselného datového typu. Generovaný SK pak často nese časovou informaci, jež urychluje různé třídění či vyhledávání (např.: prvních 6 čísel jsou kombinací roku a měsíce).

Navýšení dotazovacího výkonu je dosaženo i v případech, kdy SK nahrazuje klíče složené. Obecně platí, že s čím většího počtu sloupců je složený klíč tvořen, rychlost dotazování klesá. Proto je výhodně nahradit složený klíč nově vygenerovaným SK.

Generování SK již dle používaného českého názvu *primární klíč dimenze*, především doménou dimenzí, ovšem jeho použití ve faktové tabulce není vyloučeno. Jednoduše řečeno, u faktových tabulek jsou SK někdy generovány a někdy ne. Není toho tak potřeba, protože faktové tabulky jsou napojeny na dimenze. Tedy jednoznačné určeni faktového záznamu je dáno souborem klíčů do dimenze, které tento faktový záznam musí obsahovat.

Příprava a překlad klíčů ve faktových tabulkách, tedy nahrazení PK generovanými SK je součástí kroku transformace, avšak aby se výsledek projevil, je třeba data nahrát do předpřipravených struktur DS.

2.3.3 Nahrání dat

S každým novým příchozím záznamem do dimenzionální tabulky z OS je třeba provést porovnání PK se záznamy, jež jsou již v dimenzi obsaženy. Pokud daný záznam již existuje, je často přeskakován. Pokud ještě neexistuje, pak je uložen jako nový záznam v dimenzi a je mu vygenerován SK, zároveň dochází k překladu cizího klíče faktové tabulky, jež byl původně ve vztahu s PK dimenze tak, aby nedošlo k odstranění vazby mezi tabulkami.

Obecně, pokud přijde záznam z OS do dimenzionální tabulky, je způsob, jakým je v ní řešena historizace, určující pro rozhodování jak s tímto záznamem naložit.

Možností je několik. Pokud již záznam v dimenzionální tabulce existuje, může být příchozí záznam ignorován úplně, nebo porovnáván v každém atributu a při nesouladech je starý záznam aktualizován, eventuelně je přesán rovnou bez porovnávání s tím, že je riskován přepis stejných hodnot a tím i zatěžován výpočetní výkon. Nebo je starému záznamu uzavřena

platnost a zároveň je vytvořen záznam nový. Dokonce muže nastat případ, kdy je kontrolováno, zda daný záznam přišel, a pokud ne, tak je uzavřena platnost záznamu starého. Vše záleží na tom, jak je uchovávání historie nastaveno.

Pro krok nahrávání jsou určující pravidla mapování. Ta určují do jakých atributů v tabulkách DS budou po kroku transformace nahrány atributy tabulky z dočasného úložiště dat. Nahrání dat do struktur DS je tedy posledním krokem procesu ETL. Data skladovaná v DS slouží dále potřebám společnosti a v rámci BI jsou využívány především pro reporting a analýzy.

2.4 Dočasné úložiště dat

Dočasné úložiště dat (DSA[15]) je úložištěm, kde jsou extrakcí nahrána data z primárních systémů před jejich transformačním zpracování (13). Slouží tedy k prvotnímu, časově omezenému, ukládání netransformovaných dat. Data z primárních systémů jsou do DSA ukládány v poměru 1:1, je tedy přenesen jejich úplný obraz. Účelem je možnost transformace uložených dat bez zatížení primárních systémů. Zároveň je výchozím místem těchto transformovaných dat při kroku nahrávání do struktur DS.

After the data warehouse is designed, the next step is to design and build the interfaces between the system of record-in the operational environment-and the data warehouses.

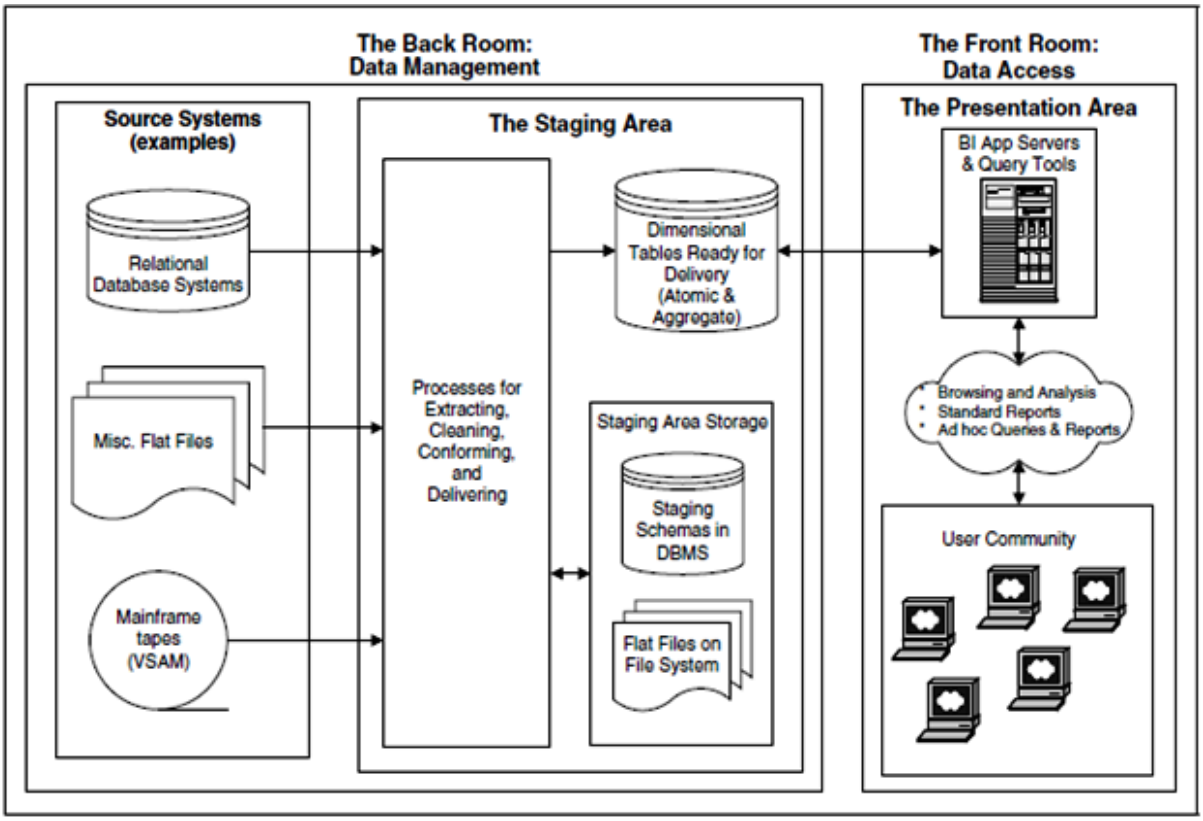
(Inmon, 2002, s. 306)

Než je taková data možno dále zpracovávat, tedy provádět nad nimi analýzy či budovat reporty, je třeba, aby bylo provedeno několik procedur, které připraví a zajistí dostupnost dat firemním uživatelům a analytikům, jež s daty manipulují. Soubor těchto přípravných procedur Kimball (2004) souhrnně nazývá *back room*, či zázemí. Back room je místem přístupným pouze zkušeným profesionálům se znalostmi datové integrace a koncový uživatel zde nemá přístup. Je zároveň místem, kam jsou data uložena po extrakci ze zdrojových systémů, očištěna, transformována a připravena k nahrání do prezentační vrstvy DS. Jakákoliv metadata generována procesem ETL, mající pro zákazníka vypovídající hodnotu, je potřeba z back room nabídnout v prezentační vrstvě DS. Back room je tedy možno definovat jako souhrnné označení pro DSA a ETL. Připravená data, vycházející z back room, jsou pak dostupná k následnému zpracování ve *front room*, jež odpovídá prezentační vrstvě DS. Back room i front room fyzicky i logicky odděleny (11).

[15] DSA – Data Staging Area

Účelem ETL systému je dodávat prezentační vrstvě dimenzionálně modelované tabulky, které jsou přímo přístupné dotazovacími a reportovacími nástroji, dashboardy a OLAP kostkami. Data ve front room jsou tedy tím, co již koncový uživatel vidí a k čemu má přístup.

Infrastruktura dle Kimballa odděluje data a procesy, které tvoří DW / BI systém do prostředí back a front room, jak je znázorněno na obr. 9.



Obr. 9 - Back room a front room, zdroj: (11)

Existuje srovnání Kimballovy infrastruktury s kuchyní v restauraci. Tak jako je v kuchyni při přípravě jídla potřeba dodržovat jisté kroky, je možné je analogicky srovnat s takovými procesy při zpracovávání dat. Zákazník restaurace pak přístup do kuchyně nemá a nemůže ani nijak ovlivnit samotnou přípravu jídla.

„Pro vysvětlení je třeba si zákazníky restaurace představit jakožto koncové uživatele a jídlo jakožto data. Jídlo je zákazníkům servírováno tak, jak si jej objednali, tedy je důležité, aby splňovalo jejich očekávání. Očištěné suroviny, organizovaná příprava a samotná prezentace jídla by měla být taková, aby každá jeho součást byla jednoduše identifikovatelná a stravitelná. Před servírováním jídla v jídelní části, je jídlo připravováno v kuchyni pod dozorem zkušeného šéfkuchaře. V kuchyni je jídlo pečlivě vybráno, očištěno, nakrájeno, uvařeno a připraveno k servírování, čili prezentaci. Kuchyně je tedy pracovním prostorem, do

kterého zákazníci nemají možnost zasahovat. V nejlepších restauracích je pak kuchyně naprosto skryta před zákazníky, kdy jsou veškeré kroky přípravy skryty jejich zraku. Případný pohled na jídlo ještě během přípravy by mohl ovlivnit výsledný dojem zákazníka při samotném stravování. Avšak pokud si zákazníci vyžádají informace o přípravě jejich jídla, samotný šéfkuchař musí vyjít z kuchyně a se zákazníkem se setkat v prostorech jídelní části, jež je pro zákazníka bezpečným, čistým a pohodlným prostředím, kde mu vysvětlí proces přípravy jídla.“

Data ze zdrojových systémů se do DSA nahrávají dvojím způsobem. Buď je nahrán kompletně celý obraz dat primárních systémů, jež je využíváno při inicializačním nahrání, nebo jsou přírůstkově nahrána data, jež od posledního nahrání v primárním systému přibyla.

Při iniciálním nahrávání do DSA, jsou data přenesena doslova do posledního znaku, neboli v DSA vznikne obraz dat 1:1. Tento proces je docela pomalý a vysoce náročný na výkon hardwaru. Vzhledem k velikosti obsahu dat ve srovnání s přírůstkovým nahráním se jedná o časově nejnáročnější typ nahrávání dat do DSA. Data jsou nahrávány i s doplňujícími logickými informacemi, např.: časovou značkou (timestamp).

V případě, kdy jsou nahrávána všechna data z primárních systémů, avšak nahrání předtím již proběhlo, nejedná se tedy o inicializační nahrání, je používán pojem plné nahrání. Tento typ nahrávání je však již časově méně náročný, vzhledem k tomu, že není potřeba udržovat žádné logické informace o změnách v nahrávaných datech od doby posledního úspěšného nahrání, jako je tomu v případě nahrání přírůstkového, ani není potřeba zavádět časové údaje, tedy timestamp, jako v případě nahrání prvotního.

V pravidelných intervalech jsou pak nahrávána dávkově data nová. Tento proces je již mnohem rychlejší a ne tolik výkonnostně náročný.

Typy nahrávání dat lze tedy rozdělit na následující:

- iniciální,
- plné,
- a inkrementální.

Vhodné je ještě zmínit real-time replikaci dat. Real-time či online replikace dat (CDC[16]) je typ nahrávání dat, jež je nejčastěji využíván na operativní úrovni řízení, zejména pak u operativních úložišť dat (ODS), viz obr. 2 s. 13. Přírůstky nejsou nahrávány dávkově, v typicky denních intervalech, ale rovnou v okamžiku, kdy jsou tato data zapsána do

[16] CDC – Change data capture

primárního systému. Jakákoliv zaznamenaná změna, či datový přírůstek je okamžitě zachycen ETL nástrojem a automaticky replikován do struktur DS. Prostoje mezi vznikem změny a momentem, kdy je změna viditelná v systému koncovému uživateli, jsou minimální.

Nahrávání dat je tedy prováděno přímým (online) či nepřímým (offline) způsobem. V případě online nahrávání je zajištěno přímé spojení mezi DSA a primárními systémy. Oproti tomu offline nahrávání je zajištěno bez přímého spojení. Data jsou v tomto případě čerpána z předem vyexportovaných datových souborů a jsou již většinou strukturovaná. Mezi takové datové soubory pak patří XML, CSV či log soubory.

2.5 Shrnutí

Teoretická část bakalářské práce čtenáři přibližuje BI jako celek, jež staví na principech datových úložišť, především DS a DSA. Pozornost je věnována vysvětlení přístupů budování DS a následnému nahrání dat do jejich struktur.

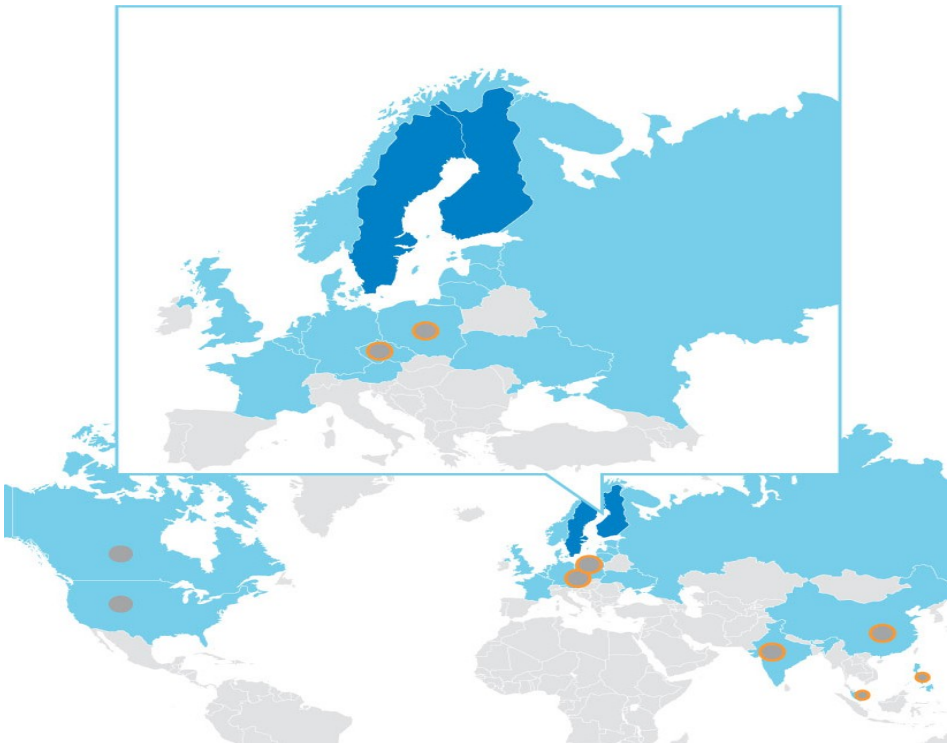
V praktické části je na základě znalosti teoretických východisek navržen a realizován jeden z přístupů budování DS a následné nahrání dat z primárních systémů pomocí procesu ETL.

3 Analýza využití technologie ETL v dané oblasti

V následující kapitole je čtenář seznámen se zadáním projektu, jež je z části realizován jako součást praktického výstupu tématu této bakalářské práce. Uveden je stručný popis společnosti, v rámci které byl projekt realizován, definice realizovaného projektu a následná analýza dat, jež byla zákaznickou společností poskytnuta. Konečnému návrhu řešení a praktické realizaci se věnuje následující kapitola č. 4.

3.1 Dodavatelská společnost

Projekt, jakožto komplexní celek, byl k vypracování zadán společnosti Tieto, jež se mimo jiné orientuje i na vyvíjení zákaznických BI řešení. Tieto je momentálně největší severoevropský dodavatel IT služeb, poskytující komplexní služby v oblasti IT pro soukromý i veřejný sektor. Zákazníkům jsou tedy nabízena řešení z oblastí podnikových systémů



Obr. 10 - Přehled největších poboček společnosti, zdroj: www.tieto.com

a portálů, integrace obchodních procesů, aplikačního outsourcingu, business inteligence, vývoje a správy aplikací, IT konzultace a další. Tieto zároveň patří mezi pět největších společností v oboru informačních technologií v České republice. Výseč největších poboček společnosti je vyobrazena na obr. 10. Celkový počet zaměstnanců, rozložených v 25 zemích,

se blíží ke 20 000. Z hlediska počtu kmenových zaměstnanců je česká pobočka třetí největší pobočkou společnosti na světě. První dvě místa zaujímají mateřské země Finsko a Švédsko. V České republice pracuje kolem 2000 zaměstnanců, z nichž většina spadá pod pracoviště v Ostravě.

3.2 Zadání projektu

Zákaznická společnost, čili zadavatel, má zájem o modernizaci datové základny v rámci všech provozovaných poboček. Jedná se tedy o celopodnikové řešení. Jedním z požadavků je integrace veškerých podnikových dat v rámci jediného datového úložiště, ke kterému bude umožněno přistupovat operativně s možností real-time analýzy uchovávaných dat. Sféra působení zákaznické společnosti (dále jen jako zákazník) se zabývá zprostředkováním finančních služeb bankovním institucím. Veškerá data, kterými zákazník disponuje, odpovídají souhrnu údajů ohledně bankovních účtů a provedených transakcí. Pod zákazníka takto spadá několik bankovních institucí.

V rámci modernizace přístupu skladování dat, je přistoupeno k návrhu budování nového DS, jež bude uchovávat veškerá data společnosti.

Nejsou však poskytnuta všechna data. Jedná se tedy pouze o určitou výseč dat, na jejíž bázi je třeba navrhnout řešení, vhodné pro komplexní použití.

Zákazník má možnost konečné BI řešení prohlížet a dále upravovat pouze, pokud sám vlastní instalaci a licenci daného BI nástroje. BI řešení tedy nelze podat v komplexním samo rozbalovacím souboru, jež by bylo nejen samostatně spustitelné, ale i schopno vykazovat relevantní výsledky či datové analýzy. BI nástroje je důležité chápat jako vrstvu, jež obsahuje OLAP kostky a je tedy jakýmsi prostředníkem mezi uživatelem a datovou vrstvou.

Zákazník musí v rozpočtu na tento projekt počítat s často nákladnou licencí BI nástroje, jež bude k realizaci využit. Licence je zároveň třeba s pravidelností jednoho roku znovu prodlužovat.

3.3 Dostupná data

O primárních systémech, ze kterých data pocházejí, nebyly poskytnuty žádné informace. Znalost typu primárních systémů zákazníka je však pro účely této bakalářské práce irelevantní. Zákazníkem byla poskytnuta vyexportovaná data typu Flat file, konkrétněji v datových souborech CSV a XLS[17]. Zároveň byla ve specifikaci požadavků projektu textově

[17] XLS – přípona souborů aplikace Microsoft Excel

doplněna o popisy a vysvětlivky. Na bázi těchto poznámek byly doplněny některé dimenzionální atributy.

Celkově se jedná o údaje ohledně kontraktů či bankovních transakcí. K dispozici je sedm CSV souborů, jež obsahují data dimenzionálního charakteru a XLS soubor obsahující záznamy jednotlivých transakcí. Struktura XLS souboru, v němž se nacházejí atributy jako typ kontraktu, peněžní částka aj., napovídá, že se jedná o tabulku faktovou. Zároveň jsou obsaženy atributy, jež se odkazují do jednotlivých dimenzí.

Přehled poskytnutých údajů s popisem a možností využití *(D - dimenze, F- fakta)*.

Název tabulky	**Flat file**	**Typ**	**Počet atributů**	**Počet záznamů**
Contract	XLS	F, D	31	1000
Currency	CSV	D	11	307
Industry	CSV	D	11	1912
Instrument	CSV	D	11	59
Intent	CSV	D	11	202
Interest rate	CSV	D	11	1991
Sector	CSV	D	11	3190
Transfer	CSV	D	11	60

Tabulka 1 - Přehled zdrojových dat, zdroj: autor

ETL zpracovávající tato data může vycházet rovnou z tabulek jakožto zdroje dat, avšak k účelu organizovanosti a jednodušší manipulace je vhodnější tyto tabulky nejprve importovat do DSA, jež bude dále sloužit jako zmiňovaný zdroj dat.

Poskytnuté tabulky však nejsou dostačující. Pro správnou funkčnost a dodržení BI pravidel je třeba dále vytvořit či vygenerovat tabulky, jež doplní ty stávající tak, aby jejich výsledné propojení v rámci DS fungovalo jako komplexní celek.

3.3.1 Classification

Nejjednodušší dimenzí, jež je třeba v DS vytvořit, je klasifikace jednotlivých údajů, vycházející z atributu Clasification ID. Tento atribut se nachází ve většině dimenzí a určuje, na jaké úrovni agregace se daný záznam nachází a pro jaký typ reportu je určen. K vytvoření nové dimenze postačí data distingovaná, jež budou doplněna o atribut s daty, vycházejících z doprovodných poznámek, v tomto případě názvů jednotlivých typů reportů.

3.3.2 Sector group

Podobným způsobem je možno postupovat při tvorbě dimenze, jež seskupuje v hierarchii nejvýše postavené sektory. I přesto, že všechny poskytnuté dimenzionální tabulky obsahují atribut, jenž se odkazuje na rodičovský záznam (*parent*) dané dimenze, tedy existuje vždy určitá hierarchie, využitelnost je v rámci této bakalářské práce minimální. Vzhledem k tomu, že výsledný DS je doslova několikanásobně větší, co se týče do obsahu i množství tabulek, není možné správně využít všechny tabulkové atributy. Původní řešení obsahuje tzv. *bridge* tabulky, jež slouží k účelu zachování hierarchie záznamů. Tyto tabulky propojují dimenzi a faktovou tabulku.

V rámci této bakalářské práce je vhodné využít pouze jedno hierarchické rozdělení *parent-child,* a to u dimenze Sector. Jednotlivé sektory totiž spadají pod organizačně vyšší celky, tedy jakési nad-sektory (*sector group*). Faktová tabulka obsahuje atribut SectorCD (*customer sector group)*, jež odkazuje na jednotlivé organizační celky.Je tedy třeba z tabulky Sector distingovat v hierarchii nejvýše postavené záznamy a ty pak vložit do nově vytvořené dimenze D_Sector_Group.

3.3.3 Day

Faktová tabulka Contract obsahuje atribut datum. Je možné předpokládat, že datum, v jakémkoliv formátu, od určení dne v týdnu po zjištění přestupného roku, bude v DS nějakým způsobem dále využíváno. Je tedy vhodné vygenerovat dimenzi, jež bude obsahovat veškeré údaje typu datum v určeném rozmezí několika let. Obsaženy budou atributy jako datum, den v týdnu, den v roce, měsíc, přestupný rok aj.

3.3.4 Country

Ve faktové tabulce je zároveň obsažen atribut s ISO kódy zemí. Pro úplnost dat je vhodné, aby existovala dimenze samostatná, jež bude kromě ISO kódu země obsahovat i její plný název. Taková dimenze pak bude propojena s tabulkou faktů. Seznam ISO kódu odpovídá směrnici ISO 3166-2, tedy dvouznakové zkratky zemí, již možno dohledat na oficiálních stránkách ISO.org.

3.3.5 Amount type

Stejně jako dimenze Classification, vychází atributy dimenze Amount type ze zákazníkem poskytnutých poznámek. V atributech jsou obsažena data ohledně typu platby

a použitých peněžních prostředků s doplňujícími údaji ohledně kategorie, do které taková platba spadá.

Všechny dimenze by měly mít zároveň genericky vytvořen SK, jež bude nejen jedinečným identifikátorem záznamu tak jako PK, ale zároveň je vhodnější jej použít při propojení dimenzionální tabulky s tabulkou faktů.

3.4 Využití ETL nástroje

Všechna data ze všech poskytnutých datových souborů budou importována do DSA. DSA bude sloužit jako zdroj dat pro nástroj ETL, jež bude použit k návrhu transformací nad danými daty a pro jejich následné nahrání do struktur DS.

Jak DSA tak DS budou uloženy jako schéma společné databáze, jež bude vytvořena v MSSQL. Databáze je uchovávána na lokálním serveru.

Vzhledem k tomu, že data v DSA nebudou muset být v průběhu přesunu ze zdrojových souborů jakkoliv modifikována, je možno využít integrovaného nástroje pro import dat do databázového systému. Struktury se tak nemusí předem připravovat a návrhy tabulek budou generovány automaticky. I tento přístup má však svá úskalí. Zde se jedná o možný problém nesprávného uložení dat v nevhodném datovém typu.

Pomocí DML příkazů jazyka SQL je třeba vytvořit tabulky v DS a definovat datové typy jednotlivých atributů. Struktura tabulek DS bude generického charakteru. Tedy veškeré atributy budou souhlasné pro každou tabulku s pouhou obměnou atributů, jež jsou identifikujícího charakteru, typicky atribut klíče.

Ze zadání projektu vyplývá potřeba transformace dat. Dle kapitoly 2 je možno data transformovat způsobem dvojím. Vzhledem k charakteristice projektu, je vhodnější využít způsobu, kdy je k transformacím využit nástroj ETL, namísto kombinace příkazů OS a SQL. Avšak SQL je v nemalé míře využíván jednotlivými komponentami ETL nástrojů a obecně lze vytvořit kompletní proces v rámci těchto nástrojů pouze s využitím jazyka SQL. V takovém případě se využívá především výkonnosti a programovatelnosti, jež nástroje poskytují.

Vzhledem k partnerské spolupráci mezi společnostmi Tieto a Microsoft (MS) bude k návrhu DSA i DS i jejich následné správě, využíván MS SQL Server 2012 (MSSQL). Jako ETL nástroj bude využit SQL Server Integration Services 2010 (SSIS), jež operuje na bázi MS Visual Studia. Zároveň realizaci projektu v rámci produktů jediného poskytovatele, je

možno považovat za výhodu. Je tak zaručena kooperace mezi jednotlivými nástroji a práce ve sjednoceném uživatelské prostředí.

Mezi další nástroje, jež bylo možno v rámci dodavatelské společnosti k realizaci využít, patří Informatica, DataStage či Pentaho. Tyto nástroje se pravidelně umisťují v mnoha kategoriích tzv. Magického kvadrantu, každoročně realizovaným společností Gartner. Obr. 11

Obr. 11 - Magický kvadrant databázových systémů DS, zdroj: http://www.gartner.com/

zobrazuje situaci v kategorii *Data Warehousing Database Management Systems* k lednu 2013. Jak z obrázku vyplývá, zvolené MSSQL se nachází v části lídrů dané kategorie.

S použitím MSSQL je tedy třeba vytvořit databázi. Tu je vhodné organizovat pomocí schémat, jež budou sloužit jako DSA a DS. SSIS bude využíván k návrhu procesu ETL a bude tak mezi těmito schématy data transformovat.

Pro přehlednost jednotlivých transformací je využíváno mapovacího předpisu (DTD[18]). Ten určuje, jakým způsobem jsou mapovány atributy ze zdrojové tabulky na atributy tabulky cílové, viz tabulka 2.

Zdrojová tabulka	**Transformace**	**Cílová tabulka**
Zdrojový atribut	SQL, změna datového typu, překlad SK atd.	Cílový atribut

Tabulka 2 - Obecný mapovací předpis, zdroj: autor

[18] DTD – Detailed data description

4 Návrh a praktická realizace aplikace pro transformaci dat do datového skladu

Výstupem této kapitoly je v příslušném nástroji, tedy v SSIS, spustitelná aplikace, jež transformuje data přicházející do DSA a nahrává je dle pravidel, zdokumentovaných mapovacím předpisem, v odpovídajícím formátu, do struktur DS.

Vzhledem k požadavku vyhnutí se jakékoli kompromitaci porušení bezpečnosti údajů, byla zpracovávaná data modifikována. Způsob modifikace se lišil dle typu zpracovávaných dat, ve většině případů se pak jednalo o změnu číselných hodnot, případně pozměnění názvu či kódového označení. Modifikovaná data však realizaci ETL procesu nijak neomezovala a posloužila zcela stejně, jako data původní. Jméno společnosti zákazníka v této bakalářské práci zmíněno nebude.

První část kapitoly je věnována popisu tvorby nové databáze a importu dat do DSA. Následuje popis postupu při tvorbě aditivních tabulek, jež slouží jako základ pro dimenze DS.

```
    /*Bloky kódu SQL jsou od běžného textu rozlišeny ohraničením
odstavce a rozdílným typem písma. */
```

4.1 Import dat

Jak již bylo zmíněno v předchozí kapitole, přímý přístup k primárním systémům zákazníka zaručen nebyl. Přenos dat byl uskutečněn přes vyexportovaná data typu Flat file. Kromě poskytnutých dat byly pro úplnost koherentního celku dimenzionálního modelu vytvořeny i další tabulky, jež budou dále transformovány do dimenzí.

V nástroji MSSQL byla vytvořena databáze s názvem *BP_Derjan*. V rámci této databáze jsou využívána dvě schémata. Schéma *stage*, jež slouží jako DSA a schéma *dw*, sloužící jako DS.

```
    CREATE DATABASE BP_Derjan
    CREATE SCHEMA stage
    CREATE SCHEMA dw
```

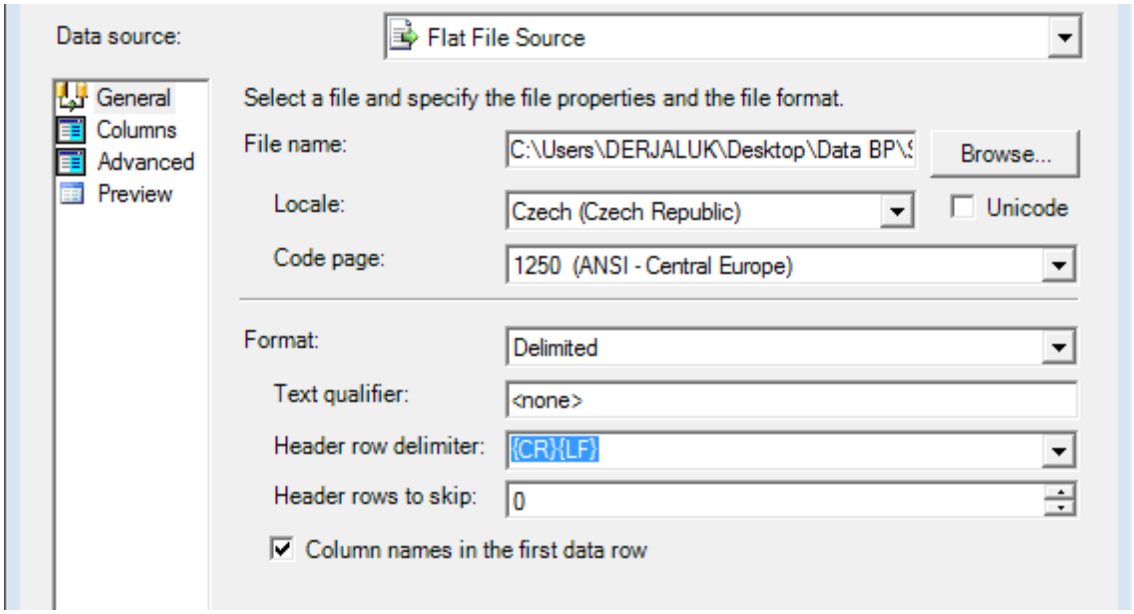
Schéma databáze plní především funkci organizačního charakteru a v případě této BP nahrazuje výchozí schéma *dbo*. Je jakýmsi kontejnerem objektů, ke kterému je možno omezit či garantovat přístupy a práva uživatelům databáze.

Data ze zdrojových souborů byla importována s využitím pomocného integrovaného nástroje SQL Server Import and Export Wizard. V rámci tohoto nástroje je možno jednoduše a přehledně importovat data z rozličných zdrojů do předem připravených struktur v MSSQL, nebo nechat tyto struktury vytvořit automaticky během importu.

Import probíhá v několika krocích. Výběr zdroje, cílové destinace a jednotlivých importovaných sloupců.

V prvním kroku je zvolen zdrojový soubor (*data source*), jež má být importován. V případě importu poskytnutých CSV souborů byla vybrána možnost Flat file, pro soubor typu XLS pak Excel, nejlépe v odpovídající verzi.

Prostředí Wizardu v prvním kroku, viz obr. 12, se skládá ze záložky pro obecný přehled informací ohledně nastavení importovaného souboru (*general*), záložky pro nastavení oddělovačů sloupců (*columns*), pokročilejších možností nastavení jednotlivých sloupců (*advanced*) a náhledu dat importovaného souboru (*preview*).



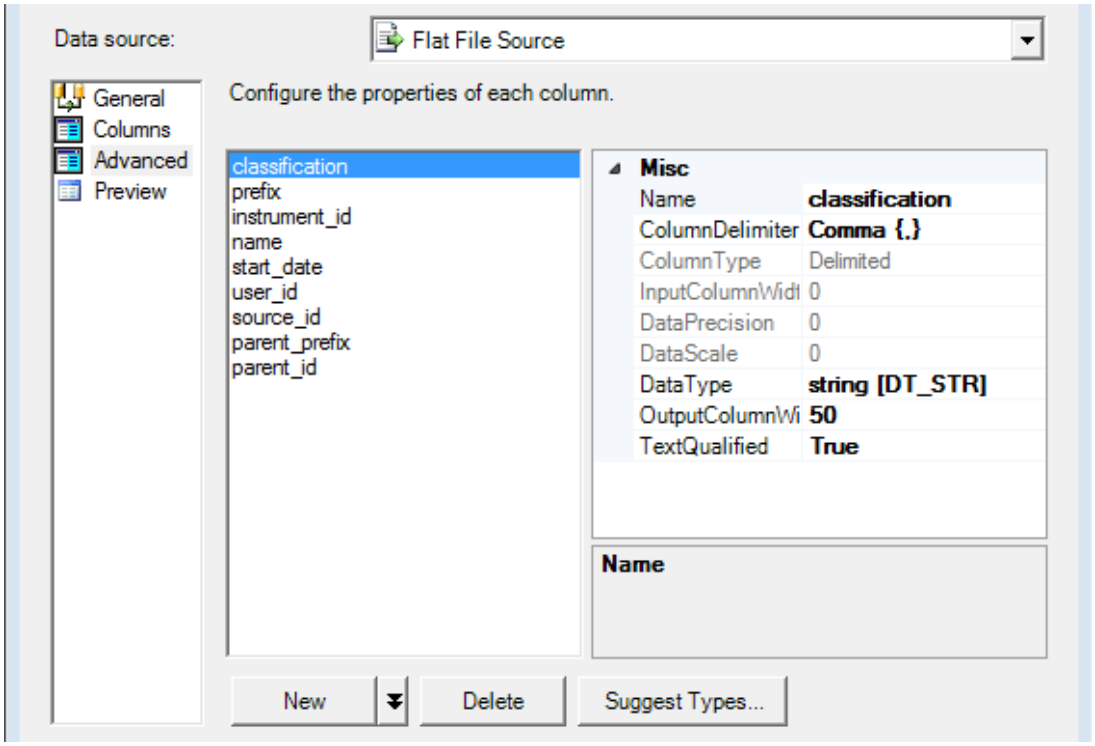
Obr. 12- Import and Export Wizard, záložka *general*, zdroj: autor

Na záložce obecné se kromě cesty k umístění importovaného souboru a případné změny kódování (*code page)* nastavuje, zdali se v tabulce nachází *záhlaví v prvním řádku*. Tato možnost je dostupná jak u souborů XLS tak u CSV. Podrobněji je možno nastavit každý ze zdrojových sloupců zvlášť na následující záložce *columns*.

Mezi relativně důležitá nastavení patří výběr vstupního datového typu, nebo nastavení oddělujícího znaku jednotlivých sloupců (*column delimiter*), obojí na záložce *advanced*. Wizard nabízí možnost navržení vhodných datových typů (*suggest types*) vzhledem k obsahu

sloupců. V případě nahrávání dat do DSA není tuto možnost třeba využívat. Vzhledem ke struktuře CSV souboru je jako oddělovač použita čárka. Nastavení je viditelné na obr. 13.

Posledním krokem importu je určení cílové destinace. Destinací může být opět datový soubor CSV, XLS, databáze (MSSQL, Access) aj. V případě realizovaného importu je destinací nativní klient MSSQL. Severem, na nějž se mají data nahrát, může být server lokální



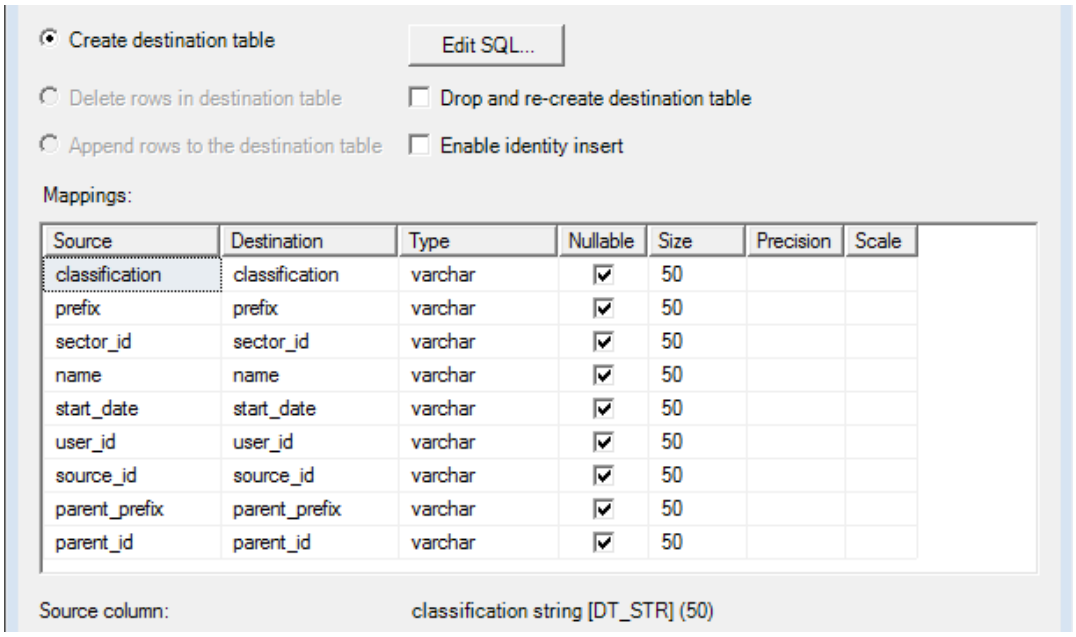
Obr. 13 – Datový zdroj importu, záložka *advanced*, zdroj: autor

(*localhost*), nebo jakýkoliv jiný server dostupný na síti. Pro přístup k serveru je potřeba správné autentizace uživatele. Autentizace je procesem ověřujícím identitu uživatele. V případě správnosti vyplněného jména serveru i přihlašovacích údajů, se uživateli zobrazí seznam dostupných databází, jež jsou na serveru obsaženy. Zároveň je možno vytvořit databázi novou.

Po dokončení obecného určení destinace importu, tedy volby cílové databáze, obsahující schémata *stage* a *dw*, je třeba podrobnějšího určení, jaká data mají být přenesena. Uživatel může napsat vlastní SQL dotaz, jež zdrojová data před konečným importem různě agreguje, seskupuje, seřazuje či jinak specifikuje jejich výběr. K dispozici je i náhled výsledku dotazu (*preview*). Pokud dotaz využit není, data jsou kompletně zkopírována do databáze.

V případě importu datového souboru XLS je možno vybrat, jaké sešity mají být importovány. Je tedy možno importovat zároveň více než pouze sešit jeden.

Zbývá pouze určit, jakým způsobem se mají sloupce importovat. Po výběru sešitu (v případě CSV je tento krok přeskočen), je možno nastavit mapování (*edit mapping*) jednotlivých sloupců, viz obr 14. Mapování je obdobné jako v případě procesu ETL, avšak v tomto případě se neprovádí jiné transformace než změna datového typu, povolení NULL



Obr. 14 - Mapování importovaných sloupců, zdroj: autor

hodnot a velikost daného atributu. Opět, pokud uživateli více vyhovuje psaní SQL příkazů než práce v GUI Wizardu, má možnost editovat automaticky generovaný SQL kód, jež vytváří tabulku v určené destinaci. Zároveň je zobrazen náhled, jaká data budou importována.

Tímto způsobem jsou importovány všechny poskytnuté datové soubory. V případě, kdy se kroky importu v rámci projektu opakují bez větších změn, je možno celý proces importu uložit a využít jako šablonu pro import příští.

Importovaná data v DSA slouží jako zdroj, odkud data čerpá proces ETL. Ten je dále dle mapovacího předpisu transformuje a nakonec nahraje do struktur DS. Tedy dříve, než se vůbec začne s návrhem ETL je třeba vytvořit DS. Avšak z profesionálního hlediska je nejvýhodnější situace, kdy je během návrhu DS brán ohled na funkčnost ETL a naopak. Jako příklad takové kooperace je možno považovat návrh generické struktury jednotlivých dimenzí DS tak, aby v průběhu procesu ETL stačila pouhá změna definované proměnné, jako třeba názvu tabulky, jež zaručí funkčnost daného procesu několika takto genericky vytvořeným dimenzím zároveň.

4.2 Struktury datového skladu

I přesto, že poskytnuté dimenze obsahují stejný počet atributů, vždy jedenáct, bylo od návrhu generické struktury ustoupeno. Rozdíly, jež se mezi některými atributy tabulek v DSA vyskytovaly, byly natolik signifikantního charakteru, jenž by znamenal podstatné změny v každé iteraci ETL procesu. Za těchto podmínek bylo přistoupeno k variantě, kdy i přes značnou podobnost mezi dimenzemi, bude každá dimenze navrhována nezávisle na ostatních a tím pádem bude pro každou vytvořen zvlášť samostatný ETL proces.

DS obsahuje 13 dimenzí. Dimenze v DS odrážejí nejen 7 importovaných tabulek z DSA, ale zároveň dimenze „uměle“ vytvořené. Dohromady pak utvářejí DS jako celek.

V následujícím textu je uživateli přiblížen postup při tvorbě využívaných dimenzí.

4.2.1 Tvorba dimenzí

Jak již bylo řečeno, struktura dimenzí, především těch, jež vycházejí z původních tabulek v DSA, viz tabulka 1, s. 37, je velmi podobná. Postup při jejich vytváření je stejný. Základem je vždy společné jádro devíti atributů. Toto jádro vychází ze struktur importovaných tabulek, viz obr. 15. Dimenze se liší pouze v atributech, jež jsou nad tímto jádrem navíc. Krom rozlišení DSA a DS s využitím schémat *stage* a *dw* v databázi *BP_Derjan*, je třeba i v rámci schéma *dw* rozlišit dimenzionální tabulky od tabulky faktové. K tomuto rozlišení je využíván prefix na začátku názvu každé z tabulek. Pro označení dimenze je použit prefix D a pro tabulku faktovou prefix F. Všechny mezery v názvu jsou zároveň nahrazeny podtržítky.

CLASSIFICATION_ID	CURRENCY_ID	PREFIX	NAME	START_DATE	USER_ID	SOURCE_ID	PARENT_PREFIX	PARENT_ID
mfi3	BGN	C	Lev, Bulgaria	2006-01-09	t1164sk010207	1	C	BGN
mfi3	BHD	C	Dinaari, Bahrain	1998-01-12	LUONTI_32	1	C	BHD
mfi3	BIF	C	Frangi, Burundi	1998-01-12	LUONTI_			
mfi3	BMD	C	Dollari, Bermuda	1998-01-12	LUONTI_			
mfi3	BND	C	Dollari, Brunei	1998-01-12	LUONTI_			
mfi3	BOB	C	Boliviano, Bolivia	1998-01-12	LUONTI_			
mfi3	BOV	NULL	MVDOL, Bolivia	2006-01-11	t1164sk			

CLASSIFICATION_ID	varchar(16)
CURRENCY_ID	varchar(16)
PREFIX	varchar(16)
NAME	varchar(255)
START_DATE	date
USER_ID	varchar(16)
SOURCE_ID	varchar(16)
PARENT_PREFIX	varchar(16)
PARENT_ID	varchar(16)

Obr. 15- Tabulka Currency a společné jádro DSA, zdroj: autor

4.2.1.1 *Společné jádro*

Mezi společné atributy navrhovaných dimenzí, vycházející z obr. 15, patří jedinečný klíč, jenž je nastaven jako identita a s každým vloženým záznamem do dimenze je automaticky inkrementován o jednotku. Tento klíč slouží pro překlad ID daného záznamu ve

faktové tabulce. Plní tedy funkci SK. Struktura jeho názvu je složena z názvu dimenze, podtržítka a slova *KEY*, např.: Transfer_KEY.

Mezi další atributy společného jádra patří ID záznamu, název záznamu, indikátor aktuálnosti, počáteční a koncová data platnosti, automaticky generovaná časová značka vložení daného záznamu do dimenze, ID zdroje a název klasifikace v rámci reportingu.

Při vytváření struktury dimenzí je využíváno příkazu CREATE TABLE jazyka SQL. Uvedený příklad ukazuje strukturu dimenze D_Currency, jež uchovává informace o měně, která byla během transakce používána.

```
CREATE TABLE [dw].[D_Currency](
      [CURRENCY_KEY] [bigint] IDENTITY(1,1) NOT NULL,
      [CURRENCY_ID] [varchar](16) NOT NULL,
      [NAME] [varchar](255) NULL,
      [SOURCE_ID] [varchar](16) NULL,
      [IS_CURRENT] [smallint] NULL,
      [START_DATE] [date] NOT NULL,
      [END_DATE] [date] NOT NULL,
      [CREATE_TS] [timestamp] NOT NULL,
      [CLASSIFICATION_KEY] [int] NULL,
 CONSTRAINT [PK_CURRENCY] PRIMARY KEY([CURRENCY_KEY] ASC)
```

Takto je dimenze připravena pro nahrání dat procesem ETL, avšak i po nahrání ještě nebude součástí DS jakožto celku. K dosáhnutí tohoto požadavku je třeba po konečném vytvoření všech dimenzí a jejich naplnění daty, dimenze mezi sebou propojit v diagramu tak, aby odpovídající PK byly navázány na FK. V klasickém DS je typické, aby faktová tabulka obsahovala většinu záznamů ID, jež jsou přeloženy na SK a zároveň plní funkci FK s propojením na PK dané dimenze.

Výše zmíněným způsobem a SQL příkazy, jsou s obměnou názvu atributů jako „*Název dimenze*“_KEY a „*Název dimenze*“_ID vytvořeny dimenze následující:

- D_Currency, obsahující informace o měnách,
- D_Industry, obsahující informace ohledně průmyslových oblastí
- D_Instrument, obsahující informace použitého nástroje,
- D_Intent, obsahující informace týkající se platebního záměru,
- D_Interest_ref_rate, obsahující informace úrokových měr,
- D_Transfer, obsahující informace ohledně transferu peněžních prostředků.

4.2.1.2 *Dimenze Sector a Sector group*

Pokud je srovnáno společné jádro dimenzí s atributy, jež byly tabulkám společné v DSA, viz obr. 15, s. 46, lze si povšimnout, že je ignorován atribut rodičovského záznamu i s prefixy. Bohužel nebyly poskytnuty doplňující tabulky, poznámky či datové soubory, na nichž by bylo možno vytvořit tyto dimenze v hierarchické struktuře. V případě dimenze Sector je však situace jiná. Zde je možno vycházet pouze z dostupných atributů a tak dát za vznik rozšířené dimenzi D_Sector, jež je zároveň organizována dle odvozené dimenze D_Sector_group.

```
CREATE TABLE [dw].[D_Sector](
-- zde předchází jádro se společnými atributy
      [SECTOR_GROUP_KEY] [int] NOT NULL,
      [PREFIX] [varchar](16) NULL,
CONSTRAINT [PK_D_Sector] PRIMARY KEY [SECTOR_KEY] ASC)
```

D_Sector je poslední dimenzí postavenou na společném jádře. Atributy Prefix, jež slouží k rozlišení typu skupiny, do níž daný sektor spadá a Sector_group_KEY, odkazující se na rodičovský záznam, čili sektor pod který spadá, tvoří hierarchii všech sektorů obsažených v DS. Na bázi těchto atributů je vytvořena dimenze D_Sector_group, jež sdružuje všechny rodičovské sektory pohromadě. Tyto rodičovské sektory lze získat distingovaným výběrem z atributu Sector_group_KEY. Každému z nich je vygenerován vlastní SK, jež bude sloužit k překladu původní hodnoty zmíněného atributu.

```
CREATE TABLE [dw].[D_Sector_group](
      [SECTOR_GROUP_ID] [int] IDENTITY(1,1) NOT NULL,
      [PARENT_SECTOR] [varchar](16) NULL,
 CONSTRAINT [PK_D_Sector_Group] PRIMARY KEY
([SECTOR_GROUP_ID] ASC)
```

Krom spojení této dimenze s dimenzí D_Sector přes atribut SK, je možno D_Sector_group propojit přímo i s tabulkou faktovou, jež v sobě obsahuje atribut o skupině sektoru protistrany. Tohoto je využito k rychlejší odezvě v případě dotazu na daný záznam faktové tabulky. V případě 1000 záznamů, jež má poskytnutá faktová tabulka, je rozdíl mezi přímým a nepřímým propojením zanedbatelný, avšak v případě, kdy se jedná o množství v řádu desítek tisíc, ocení uživatel rychlejší odezvu až o několik vteřin.

4.2.1.3 *Dimenze Amount type*

Jednou z dimenzí, jež vychází z poznámek zákazníka, je D_Amount_type. Ta nese informace o typu platby. V rámci poskytnutých dat existuje třicet různých typů plateb. Ty se dělí do dvou kategorií a každý typ platby je navíc speciálním znakem, jež figuruje jako určitý ukazatel výhodnosti dané platby.

```
CREATE TABLE [dw].[D_Amount_type](
      [AMOUNT_TYPE_KEY] [int] NOT NULL,
      [AMOUNT_CATEGORY] [varchar](16) NULL,
      [SIGN] [int] NULL,
      [NAME] [varchar](50) NULL,
 CONSTRAINT [PK_D_Amount_type] PRIMARY KEY
([AMOUNT_TYPE_KEY] ASC)
```

Dimenze již neobsahuje společné jádro. Atribut Amount_type_key, jež je PK dané dimenze slouží i jako SK pro překlad atributu v tabulce faktové. Na tomto atributu je pak realizováno spojení.

Poskytnuta byla data vyobrazena v tabulce 3. Pro jejich vložení do struktur dimenze není třeba procesu ETL. Je možno hodnoty vložit přímo v MSSQL, nebo využít SQL příkaz INSERT INTO.

AMOUNT_TYPE_KEY	AMOUNT_CATEGORY	SIGN	NAME
14	Money	1	Interest received
21	Money	-1	Realized credit loss
26	Percent	1	Agreed anual percentage

Tabulka 3 - D_Amount_type, zdroj: autor

4.2.1.4 *Dimenze Classification*

Další dimenze, postavená na poskytnutých poznámkách zákazníka, nabízí přehled jednotlivých klasifikačních kategorií daných záznamů. Totiž každý záznam, obsažen v původních dimenzionálních tabulkách, vlastní hodnotu atributu Classification, viz obr. 15 s. 45. Ta určuje, na jaké úrovni agregace se daný záznam nachází, a tedy pro jaký report je určen. V případě této bakalářské práce není třeba klasifikaci dále rozvádět, avšak pro potřeby reportingu je tento atribut velmi důležitý. Díky rozdělení dle typu klasifikace je vždy vytvořen zvláštní datový zdroj, tzv. dataset, nad nímž je vždy stavěn určitý report. Tímto způsobem je šetřen výpočetní výkon a zvyšována odezva reportingu. Záznamy ohledně typu klasifikace jsou uloženy v nově vzniklé dimenzi D_Classification. Ta krom atributů SK a ID obsahuje

i časovou značku *create_ts*. Ta slouží pro potřeby archivace a zaznamenává, kdy byl každý záznam v dané dimenzi vytvořen.

```
CREATE TABLE [dw].[D_Classification](
      [CLASSIFICATION_KEY] [int] NOT NULL,
      [CLASSIFICATION_ID] [varchar](10) NOT NULL,
      [CREATE_TS] [timestamp] NULL,
 CONSTRAINT [PK_D_Classification] PRIMARY KEY
([CLASSIFICATION_KEY] ASC)
```

Údaje je opět do dimenze možno vložit rovnou. Buď jejím editováním v MSSQL, nebo SQL příkazem INSERT INTO. Záznamy jsou pouze čtyři. ETL v tomto případě není potřeba.

CLASSIFICATION_KEY	CLASSIFICATION_ID	CREATE_TS
1	vira	Timestamp
2	finrep	Timestamp
3	mfi3	Timestamp
4	all	Timestamp

Tabulka 4 - D_Classification, zdroj: autor

Tato dimenze je vzhledem k množství FK, jež při propojení ostatních dimenzí s D_Classification vznikají, ve schématu DS druhou tabulkou s největším počtem vazeb, hned po tabulce faktové.

4.2.1.5 *Dimenze Country*

Faktová tabulka obsahuje atribut s dvoumístnými ISO kódy zemí, jež určují destinaci provedení dané transakce. Je vhodné dát za vznik nové dimenzi, jež bude tyto ISO kódy obsahovat a zároveň je doplní o plný název země. Tyto kódy i s názvy zemí dohledat jsou obsaženy v normě ISO 3166-2 (18). Zkopírované údaje byly importovány do schématu *stage* databáze BP_Derjan. Zároveň ve schématu *dw* byla vytvořena struktura dimenze D_Country s atributem, jež je identitou a zároveň bude sloužit jako SK.

```
CREATE TABLE [dw].[D_Country](
      [COUNTRY_ID] [bigint] IDENTITY(1,1) NOT NULL,
      [CODE] [varchar](10) NOT NULL,
      [NAME] [varchar](255) NOT NULL,
 CONSTRAINT [PK_D_Country] PRIMARY KEY
([COUNTRY_ID] ASC)
```

ETL procesem bude zaručeno naplnění dimenze D_Country daty z DSA. Zároveň se bude generovat hodnota atributu Country_ID. ISO směrnice je aktuální ke dni 6.2.2013.

4.2.1.6 *Dimenze Day*

Dimenzí, jež bude uchovávat veškeré údaje k určitému datu v daném časovém rozmezí je D_Day. Ta bude přímo napojena na tabulku faktovou a umožní tak uživateli kromě rozloženého data do dnů, týdnů a měsíců také zjistit, zdali se jedná o přestupný rok, či jaké je jméno daného dne.

Struktura dimenze D_Day v SQL:

```
Create Table dw.D_Time (
    Dateid int IDENTITY (1,1) PRIMARY KEY CLUSTERED,
    Date date,
    DateString varchar(10),
    Day int,
    DayofYear int,
    DayofWeek int,
    DayofWeekName varchar(10),
    Week int,
    Month int,
    MonthName varchar(10),
    Quarter int,
    Year int,
    IsWeekend bit,
    IsLeapYear bit )
```

Dimenze je nejen vytvořena, ale i naplněna daty vygenerovanými pomocí T-SQL. Transact SQL, neboli T-SQL je procedurální jazyk společnosti MS, jež staví na základech standardu SQL.

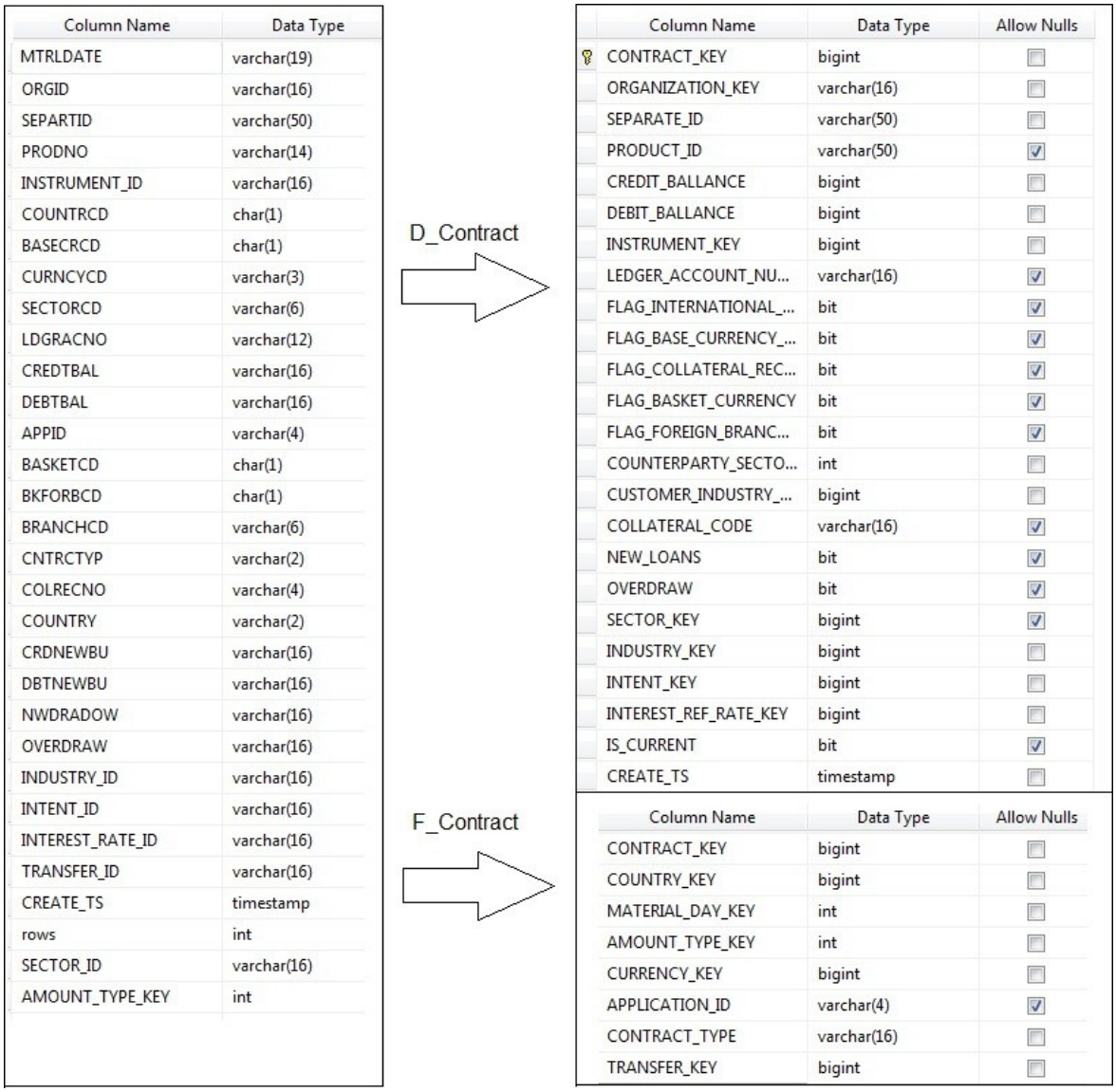
Data jsou generována iteračně. K tomuto je potřeba deklarace několika proměnných, jež v průběhu několika iterací projdou veškerá data v daném časovém intervalu. V tomto případě bylo zvoleno rozmezí mezi lety 2000 až 2020. Takto jsou pokryty veškeré dny, v rámci kterých byla data poskytnuta a zároveň je zaručena značná rezerva do budoucna. V případě, kdy by byla implementována dimenze, jež by obsahovala např. data splatnosti či jiné údaje ohledně předpokládaného časového trvání smluv, by mohla nastat situace, kdy by dané datum obsaženo nebylo, např. smlouva končící až v roce 2025, proto je třeba generovat data s rezervou.

```
Declare
@StartDate datetime,
@EndDate datetime,
@Date datetime
Set @StartDate = '2000/01/01'
Set @EndDate = '2020/12/31'
Set @Date = @StartDate
WHILE @Date <=@EndDate
BEGIN
    DECLARE @IsLeapYear BIT
    IF ((Year(@Date) % 4 = 0)
AND (Year(@Date) % 100 != 0 OR Year(@Date) % 400 = 0))
    BEGIN SELECT @IsLeapYear = 1
    END ELSE
    BEGIN SELECT @IsLeapYear = 0
    END
    DECLARE @IsWeekend BIT
    IF (DATEPART(dw, @Date) = 1 OR DATEPART(dw,@Date) = 7)
    BEGIN SELECT @IsWeekend = 1
    END ELSE
    BEGIN SELECT @IsWeekend = 0
    END
    INSERT Into dw.D_Time (
    [Date],[DateString],[Day],[DayofYear],[DayofWeek],
    [Dayofweekname],[Week],[Month],[MonthName],[Quarter],
    [Year],[IsWeekend],[IsLeapYear] )
    Values (
    @Date, CONVERT(varchar(10), @Date, 105),
    Day(@Date), DATEPART(dy, @Date), DATEPART(dw, @Date),
    DATENAME(dw, @Date), DATEPART(wk, @Date),
    DATEPART(mm, @Date), DATENAME(mm, @Date),
    DATENAME(qq, @Date), Year(@Date),
    @IsWeekend, @IsLeapYear )
    Set @Date = @Date + 1
END
```

4.2.2 Rozložení tabulky faktů

Dimenze, jež nejsou postaveny na společném jádře, nejsou ani přímo napojeny na tabulku faktů. Nastalá situace však není žádným pravidlem. Rozdělení, kdy dimenze se společným jádrem nejsou přímo napojeny na tabulku faktovou a dimenze bez společného jádra ano, vyplynulo z návrhu a konečné realizace DS.

Totiž tabulka Contract obsahuje nejen atributy, s možností překladu SK a návazností na dimenze, ale zároveň i větší množství atributů dimenzionálního charakteru, tedy takových, jež daný záznam spíše specifikují. Tabulka Contract je proto rozdělena na tabulky dvě, tabulku dimenzionální a faktovou. Struktury tabulek D_Contract a F_Contract, s datovými typy dle obr. 16, byly opět vytvořeny pomocí příkazů CREATE TABLE s PK dimenze Contract_KEY. Tento klíč realizuje spojení mezi oběma tabulkami.



Obr. 16 - Contract: dimenze i fakta, zdroj: autor

Veškeré potřebné dimenzionální i faktové tabulky v DS jsou tedy připraveny. Nyní je možno přejít konečně k návrhu a realizaci procesů ETL, jež do vytvořených struktur nahrají data z DSA. Jak již bylo zmíněno dříve, k návrhu ETL je využíváno nástroje SSIS.

Vytváření ETL ve specializovaném nástroji může čtenáři svým způsobem připomínat puzzle, kdy jednotlivé komponenty, jež jsou k sobě spojovány, zastávají rozdílnou činnost, jež je s procházejícími daty prováděna.

4.3 Komponenty SSIS

MS SSIS je jedním z nástrojů obsažených v plné instalaci MS SQL Serveru. Je volitelnou aditivní součástí programu MS Visual Studio, které zároveň slouží jako vývojové prostředí programovacích jazyků jako C#, Visual Basic a C++ i jako nástroj k tvorbě reportů.

V rámci SSIS existuje více než padesát rozdílných komponent, jež lze různě mezi sebou kombinovat ve výsledné puzzle, tedy aplikaci, jež je schopna vstupní data transformovat a nahrát do DS dle požadavků uživatele. Každá z těchto komponent v rámci své funkčnosti dále nabízí i rozšířená nastavení. V případě realizovaného projektu bylo těchto komponent využito osm.

Číselné odrážky v seznamu odpovídají číslování v obr. 17. Obrázek ilustruje ikony, zastupující jednotlivé komponenty v rámci nástroje SSIS.

Obr. 17 - Komponenty SSIS, zdroj: autor

1) OLE DB Source,
2) Union All,
3) Lookup,
4) Derived Column,
5) Data Conversion,
6) OLE DB Destination,
7) Multicast,
8) Conditional Split.

4.3.1 OLE DB Source

Komponenta zajišťující připojení na zdrojová data typu OLE DB. Typicky tedy připojení na tabulku z DSA, avšak zdrojem dat může být i pohled či SQL příkaz. Specifikovány jsou i vstupní atributy.

4.3.2 Union All

V případě dvou oddělených toků dat v ETL, např.: při dvou vstupech OLE DB Source, jsou v této komponentě data sloučena v jeden výstupní tok, jež bude obsahovat data z obou vstupních.

4.3.3 Lookup

Komponenta porovnává vstupní tok se specifikovanou tabulkou či pohledem a na bázi společných záznamů těchto dvou pak podává výstup. Takto je možno porovnávat existující záznamy v tabulce, neexistující, shodné jen v určitých atributech atd. Této komponenty se často využívá při nahrazování hodnot shodných atributů. Příkladem je již dříve zmiňovaný překlad SK. V takovém případě jsou hodnoty vybraných vstupních atribututů porovnávány s hodnotami atributů pohledu či tabulky, na níž je v rámci komponenty definováno připojení. Pokud existuje shoda mezi záznamy, je hodnota původního atributu nahrazena hodnotou atributu SK určené tabulky.

4.3.4 Derived Column

Odvození nového sloupce, či nahrazení stávajícího dle definovaného výrazu. Komponenta je využívána i v případě, kdy je třeba nahrazovat NULL hodnoty atributu za hodnoty výchozí, či na bázi podmínky definovaného výrazu dosadit určitou hodnotu atributu.

4.3.5 Data Conversion

V rámci této komponenty je možno konvertovat datový typ atributů. Konverze musí být navržena s ohledem na obsažená data tak, aby nedošlo k jejich případné ztrátě. Ztrátou se myslí volba nevhodného datového typu či délky daného datového typu tak, kdy je původní hodnota atributu tímto ovlivněna a například je ztracena část textového řetězce.

4.3.6 OLE DB Destination

Většinou zakončující komponenta, jež určuje destinaci, kam jsou data po všech operacích transformace nahrána. Typicky je destinací struktura DS, tedy předpřipravené

dimenzionální či faktové tabulky. V rámci této komponenty se nastavuje mapování vstupních atributů z datového toku do atributů dané destinace v DS.

4.3.7 Multicast

Rozdělení datového toku na totožné toky dva. Komponenta je využívána v případě, kdy je požadováno daná data zpracovávat dále dvojím způsobem.

4.3.8 Conditional Split

Opět rozdělení datového toku, avšak v tomto případě se jedná o dělení na bázi definované podmínky. Uživatel tedy může rozdělit vstupní datový tok na menší skupiny dat, jež společně splňují danou podmínku, např.: rozdělení pracovníků na muže a ženy.

Komponenty v SSIS je možno kombinovat v libovolném množství a často i nezávisle na jejich pořadí. Vždy záleží pouze na definovaném výstupu, v jakém mají být data nahrána do struktur DS.

4.4 Návrh ETL

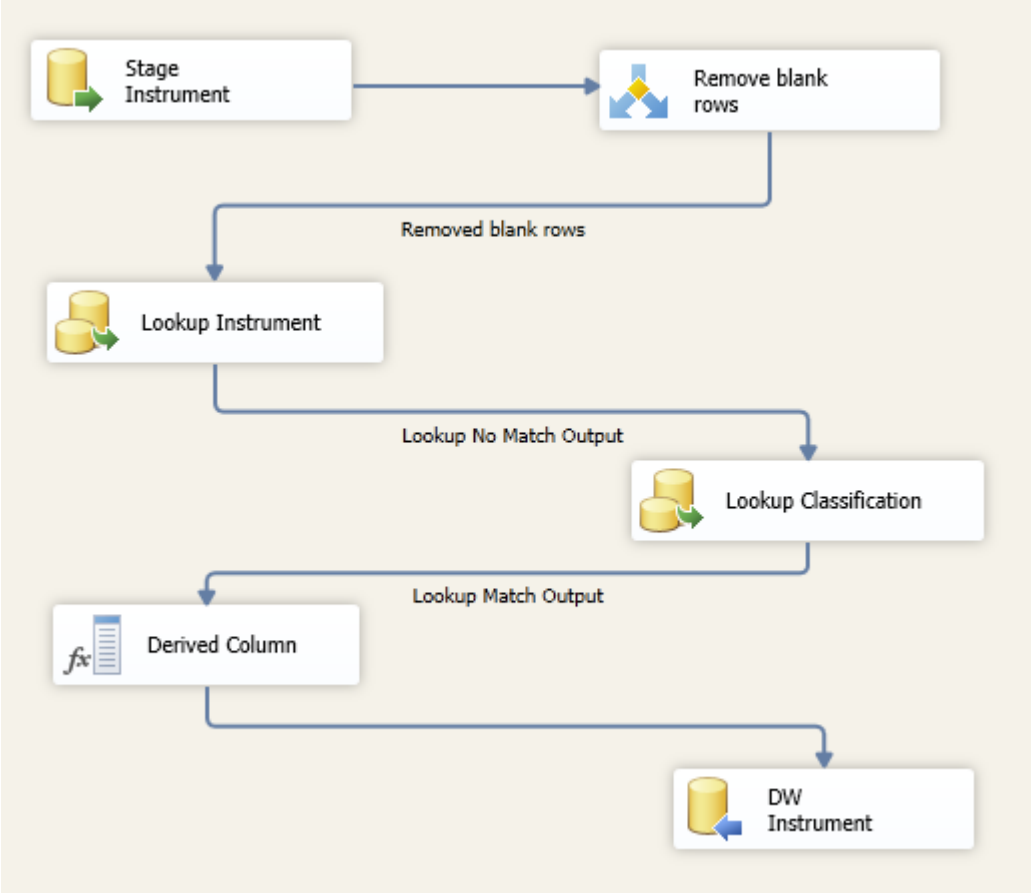
Vzhledem k tomu, že některé dimenze DS byly hned po vytvoření naplněny daty, bude se návrh procesů ETL týkat pouze těch tabulek, u kterých takto učiněno nebylo.

Při plnění DS daty je důležité dodržovat pořadí, v jakém jsou data nahrávána. Typicky jsou data nejdříve nahrávána do dimenzí, jež v sobě nenesou informace odvozené z tabulek jiných, či se na ně nějakým způsobem neodkazují. Z těchto dimenzí je poté možno odvodit číselníky. Poté je možno nahrát data do tabulky faktové. Data, jež jsou ve faktové tabulce ve výsledku nahrána, se často zdánlivě nepodobají datům původním, které tabulka obsahuje v DSA. Tento rozdíl vzniká zmiňovaným překladem atributů za SK dimenzí, jež jsou na faktovou tabulku napojeny.

Výhodou procesu ETL je jeho znovu využitelnost. V případě, kdy jsou data do dané dimenze nahrávána dávkově, třeba každý měsíc, či na konci pracovního dne, je celý proces navržen pouze poprvé. Další nahrávání pak probíhá automaticky, dle naplánovaného intervalu spouštění procesu. Období mezi koncem pracovního dne a začátkem nového, kdy se pošlou nová data primárních systémů v dávce na zpracování do DS, bývá někdy nazýváno jako business window.

Naplnění dimenzí, vytvořených na společném jádře, tedy se shodnou strukturou, je taktéž realizováno shodnými ETL procesy. Proces vždy začíná připojením na zdrojová data. Takto vznikne nový datový tok, který je možno dále transformovat. V případě, že se ve

zpracovávaných datech nacházejí NULL hodnoty atributů, jež znemožňují správnou interpretaci daného záznamu, tedy jsou v rozporu s výstupem BI, kdy je koncovému zákazníkovi důležité poskytnout informaci, jež je z daných dat získána, je takový záznam vynechán. Po té, je třeba provést kontrolu, zdali již daný záznam v dimenzi neexistuje. Existující záznamy jsou opět vynechány a dále v datovém toku pokračují pouze ty, jež v dimenzi ještě obsaženy nejsou. Vzhledem k tomu, že se jedná o inicializační nahrávání dat do struktur DS, a tedy v daných dimenzích ještě žádná data obsažena nejsou, je vstupní tok dat roven tomu výstupnímu. V dalším kroku je přeložen Classification_Key hodnotou SK dimenze D_Classification. Tímto je předcházeno redundanci dat. Nyní již zbývá jen nechat vytvořit nový atribut, jež bude určovat hodnotu aktuálnosti daného záznamu a data je možno nahrát do určených struktur dimenze. Tímto způsobem byly navrženy všechny ETL procesy společné dimenzím se společným jádrem. Výsledná podoba ETL pak odpovídá obrázku 18.



Obr. 18 - ETL D_Instrument, zdroj: autor

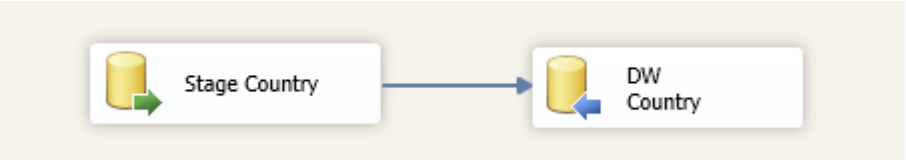
Základem je *OLE DB Source* a *OLE DB Destination*, jež ohraničují začátek a konec daného datového toku (není tomu vždy pravidlem). V komponentě OLE DB Source je nastaven přístup ke zdrojové tabulce z DSA a zároveň určeny vstupní atributy. Komponenta *Condition Split* kontroluje vstupní data pomocí podmínky, zdali jsou obsaženy hodnoty

NULL, do výstupu jsou vpuštěny pouze taková data, jež mají hodnoty atributů v pořádku. Následuje komponenta *Lookup*, jež kontroluje existenci daného záznamu v určené tabulce, v tomto případě je určena dimenze, jež je zároveň cílová. Datový tok, vystupující z této komponenty tedy osahuje data, jež neobsahují NULL hodnoty atributů a zároveň jsou jedinečná. Druhá komponenta *Lookup* je připojena na D_Classification a překládá Classification_key za hodnotu SK dané dimenze. Komponenta *Derived Column* přidává nové atributy s časovou značkou vložení záznamu a ukazatelem aktuálnosti daného záznamu. Opět vzhledem k tomu, že se jedná o inicializační nahrání, jsou všechny nahrávané záznamy určeny jakožto aktuální. V posledním kroku je určena cílová destinace, tedy dimenze, do níž se mají transformovaná data nahrát. Určena jsou i pravidla mapování, tedy přiřazení vstupních a během transformace vytvořených atributů k atributům cílové dimenze.

Nahrávání dat do dimenzí D_Country a D_Sector_group je zaručeno jednoduchým ETL procesem, obsahujícím pouze dvě komponenty; *OLE DB Source* a *OLE DB Destination*, viz obr. 19. V případě D_Country je jako zdrojový soubor vybrána importovaná tabulka obsahující ISO kódy zemí. Z této tabulky vybírá data SQL dotaz, kterým jsou ořezány případné mezery na začátku i konci řetězce každé z hodnot atributu. Tímto způsobem jsou zároveň eliminovány případné chyby vzniklé kopírováním těchto dat do DSA.

```
SELECT
LTRIM(RTRIM(CODE)) as code,
LTRIM(RTRIM([COUNTRY NAME])) as name
FROM stage.ISO_COUNTRY
```

Datový tok rovnou pokračuje do destinační tabulky, čili D_Country, v rámci které je automaticky při vkládání záznamů generován SK záznamu.

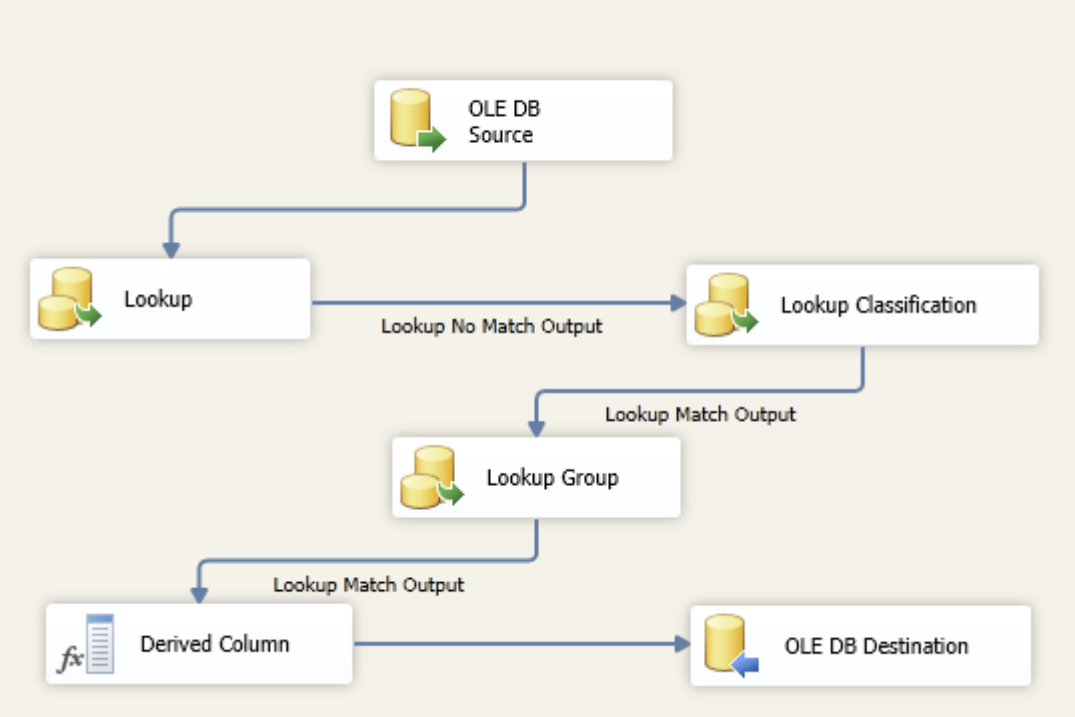


Obr. 19 - ETL D_Country & D_Sector_group, zdroj: autor

V případě dimenze D_Sector_group je v rámci komponenty OLE DB Source použit SQL dotaz, zajišťující distingovaný výběr v hierarchii nejvýše položených záznamů.

```
SELECT DISTINCT Sector_group_id
FROM stage.SECTOR
```

Obdobným způsobem, jako při naplňování dimenzí se společným jádrem je postupováno při návrhu ETL procesu pro naplnění dimenze D_Sector. Pořadí i nastavení komponent je velmi podobné tomu z obr. 18. Počet komponent je rovněž shodný, avšak komponenta *Condition Split* byla vynechána a nahrazena další komponentou *Lookup*. Ta zajišťuje, stejným způsobem jako *Lookup* do dimenze D_Classification, překlad hodnoty



Obr. 20 - ETL D_Sector, zdroj: autor

atributu Sector_group_KEY za SK dimenze D_Sector_group. Pořadí těchto komponent je vyobrazeno na obr. 20. Jak je možno si povšimnout, výstupní datový tok komponent *Lookup* se liší v označení. V případě, kdy se jedná o „Lookup No Match Output“, jsou do výstupu směrovány záznamy, jež byly nesouhlasné se záznamy porovnávané dimenze, tedy ty záznamy, jež obsaženy ještě nebyly. Naopak u výstupu „Lookup Match Output“ jsou do výstupu směřovány záznamy, jež byly v porovnávaných atributech shodné, tedy záznamy, u kterých docházelo k překladu SK.

Poslední a zároveň nejrozsáhlejší a výpočetně nejnáročnější ETL proces rozděluje tabulku Contract na tabulky dvě, dimenzionální a faktovou, viz obr. 16, s. 52. Obrázek posledního ETL procesu je vzhledem ke své velikosti přesunut do příloh, viz obr. 1, příloha č. 1, tak jako mapovací předpis, ze kterého prováděné transformace vycházejí.

První komponenta *OLE DB Source* odkazuje na zdrojovou tabulku stage.Contract. Následují *Lookup* komponenty, zajišťující na vzniklém datovém toku překlady SK. Popisek komponent *Lookup* z přílohy č. 1 koresponduje s danou dimenzí, jejíž atribut je k překladu

využit. Komponenta *Derived Column* zajišťuje transformace hodnot atributů dle mapovacího předpisu, viz příloha č.2. Následují opět dvě *Lookup* komponenty, kdy v případě kontroly záznamů v dimenzi D_Industry je zapotřebí, aby výstupní datové toky utvořily jeden tok souvislý. Dělení je zaviněno NULL hodnotami u záznamů faktové tabulky v atributu, jež se má odkazovat právě na dimenzi D_Industry. Vzniknou tak dva výstupní toky dat. Jejich spojení v tok jeden umožňuje komponenta *Union All*. Stejným způsobem je postupováno i u výstupu z následující *Lookup* komponenty do dimenze D_Industry ze strany zákazníka. Rozdělení za účelem naplnění jak tabulky dimenzionální, tak i tabulky faktové, je řešeno komponentou *Multicast*, duplikující datový tok ve dví. Tímto způsobem je jeden datový tok rovnou směrován do komponenty *OLE DB Destination*, kde jsou mapovány patřičné atributy do dimenze D_Contract. Druhý datový tok dále pokračuje v transformacích. Před tímto rozdělením jsou do atributů obsahující NULL hodnoty dosazeny dle mapovacího předpisu hodnoty výchozí, většinou nula nebo N/A. K tomuto je opět využito komponenty *Derived Column*.

Duplicitní datový tok dále pokračuje v překladech SK v rámci dalších *Lookup* komponent. Jeden z atributů datového toku nesoucí informaci datum, avšak špatného datového typu, v tomto případě text, je třeba konvertovat v datový typ správný, tedy datum. K tomuto je využito komponenty *Data Conversion*. Hodnoty atributu jsou po konverzi pomocí *Lookup* komponenty přeloženy dle Sk dimenze D_Day. Nakonec je datový tok ukončen komponentou *OLE DB Destination*, odkazující na tabulku F_Contract.

V tomto ETL procesu byly využity veškeré dříve pospané komponenty, viz obr. 17, s. 53, krom komponenty *Condition Split*.

4.5 Shrnutí řešení

Pomocí navržených ETL procesů byly naplněny všechny předem vytvořené tabulky v DS. Nyní je možné tabulky mezi sebou propojit a dát tak za vznik DS jakožto komplexnímu celku.

Po aplikování propojení FK s PK mezi dimenzemi a faktovou tabulkou vzniká diagram DS, vyobrazen na příloze č.3.

5 Zhodnocení řešení

Výsledná aplikace, vytvořena s použitím nástroje SSIS, je připravena k dalšímu používání. Je jí zajištěna extrakce dat z DSA, připraveny jednotlivé kroky transformace a konečné nahrávání do připravených struktur DS.

Jak již bylo řečeno dříve, zpracovávaná data musela být před praktickou realizací v rámci této práce modifikována. Jedná se o údaje citlivé a zákazník si nepřeje, aby byly v původní podobě zveřejněny. Aplikace tedy odpovídá realitě s lehkými změnami, jako jsou jiné názvy atributů, či jiným způsobem aplikovaná pravidla transformace kvůli rozdílnosti datového typu atributu po modifikaci. Funkčnost však plně odpovídá realitě a v originálním vyhotovení je aplikace dnes v provozu.

Transformace je možno plánovat dle časového intervalu, kdy přicházejí nová data do DSA, takto má zákazník k dispozici vždy aktuální data, tedy i reporty a analýzy, jež by nad tímto řešením byly postaveny, by automaticky měnily svůj výstup v čase a dávaly tak stále aktuální informace pro podporu rozhodování.

Řešení je tedy součástí komplexního BI celku. Data z rozličných zdrojů jsou sjednocena v rámci jednoho DS v rámci společnosti. Nejrůznější manipulace s uloženými daty jsou díky sjednocenosti ve formátu jednodušší na provedení. Taktéž možností jednoduché modifikace vytvořených procesů ETL v rámci pokračující spolupráce mezi zákaznickou a dodavatelskou společností, je zajištěna stálá podpora a aktuálnost poskytovaného řešení.

V rámci transformace bylo dáno za vznik nové faktové tabulce rozdělením té původní na tabulky dvě, kdy byla data dimenzionálního charakteru odfiltrována do nové dimenze.

Po konečném naplnění všech tabulek daty z DSA a aplikování vazeb mezi nimi, vznikla komplexní struktura nového DS, jež je nyní plně v provozu.

6 Závěr

Cílem práce bylo navrhnout a prakticky realizovat řešení pro transformaci dat z primárních systémů do struktur datového skladu. V rámci konečného řešení byla zahrnuta i celková realizace datového skladu.

V první části této práce byl čtenář seznámen s pojmem business inteligence, jakožto rozsáhlým procesem, jež mění data na informaci a informace na znalosti. Infrastruktura business inteligence vychází z datových skladů. Je vysvětlen rozdíl mezi dvěma přístupy jejich budování a to přístupem dimenzionálním, prosazovaným Ralphem Kimballem a přístupem integrovaným, prosazovaným Billem Inmonem. Dále je vysvětlen pojem dočasného úložiště dat a metody ETL, jež provádí jednotlivé kroky manipulující s daty, tedy kroky extrakce, transformace a nahrávání dat do struktur datového skladu.

Druhá část této práce byla zaměřena na analýzu využití ETL v daném problému a následnému návrhu struktur datového skladu. Návrh se řídil dle pravidel dimenzionálního modelu a výsledná struktura je formována dle schématu sněhové vločky. Importovaná data byla poté transformována a nahrána do takto připravených struktur navrženými procesy ETL.

K návrhu a správě datového skladu bylo využito nástroje Microsoft SQL Server 2012 a k návrhu aplikace, jež data transformovala a nahrávala do struktur vytvořeného datového skladu, nástroje Microsoft SQL Server Integration Services.

Výsledná aplikace splňuje dané cíle bakalářské práce a je dále využitelná k budoucímu používání. Zákaznická společnost byla s řešením spokojena.

Dané řešení spadá do odvětví business inteligence, jež na bázi takto připravených dat provádí analýzy a přehled v různých dashboardech, jež monitorují důležitá KPI, sloužící pro podporu rozhodování. Zároveň je na tuto práci možno navázat právě prací zaměřenou na data mining či reporting.

.

Seznam použité literatury

(1) LOSHIN, David. *Business intelligence: the savvy manager's guide, getting onboard with emerging IT.* Boston: Morgan Kaufmann Publishers, 2003. ISBN 15-586-0916-4.

(2) HROCH, Michal a Pavel CACH. *Business intelligence staví na datovém skladu.* SystemOnLine [online]. 2007 [cit. 2013-01-04]. Dostupné z: http://www.systemonline.cz/business-intelligence/business-intelligence-stavi-na-datovem-skladu.htm

(3) SKLENÁK, Vilém. *Data, informace, znalosti a Internet*. Praha: C.H. Beck, 2001. ISBN 80-717-9409-0.

(4) KUBÁSEK, M.; HŘEBÍČEK, J.; *Environmentální informační systémy I.* [online]. 2004 [cit. 2013-01-02]. Dostupné z: http://www.fi.muni.cz/~hrebicek/eis/EIS_1.pdf

(5) WIIG, Karl M. *Knowledge management foundations: thinking about thinking : how people and organizations create, represent, and use knowledge*. Arlington: Schema Press, 1993. ISBN 09-638-9250-9.

(6) KIMBALL, Ralph a Margy ROSS. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling.* 2nd ed. New York: Wiley, 2002. ISBN 0-471-20024-7.

(7) LACKO, Luboslav. *Databáze: datové sklady, analýza OLAP a dolování dat*. Brno: Computer Press, 2003. ISBN 80-7226-969-0.

(8) GÁLA, L., POUR, J., ŠEDIVÁ, Z.; *Podniková informatika*. 2. vyd. Praha: Grada, 2009. 496 s. ISBN 978-80-247-2615-1.

(9) NOVOTNÝ, O.; POUR, J.; SLÁNSKÝ, D.; *Business Intelligence: jak využít bohatství ve vašich datech*. Praha: Grada Publishing, 2005. ISBN 80-247-1094-3.

(10) DAVENPORT, R.J.; *ETL vs ELT: A subjective view.* [online]. 2008 [cit. 2013-02-25]. Dostupné z: http://www.dataacademy.com/files/ETL-vs-ELT-White-Paper.pdf

(11) KIMBALL, Ralph a Joe CASERTA. *The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data*. Indianapolis: Wiley, 2004. ISBN 07-645-6757-8.

(12) TVRDÍKOVÁ, M. *Aplikace moderních informačních technologií v řízení firmy: nástroje ke zvyšování kvality informačních systémů.* Praha: Grada Publishing, 2008. ISBN 978-80-247-2728-8.

(13) INMON, William H. *Building the Data Warehouse.* 4th ed. Indianapolis: Wiley, 2005. ISBN 978-0-7645-9944-6.

(14) DATE, C. *An Introduction to Database Systems*. 8th ed. Boston: Pearson/Addison-Wesley, 2003. ISBN 03-211-9784-4.

(15) KALUŽA, Jindřich a Ludmila KALUŽOVÁ. *Modelování dat v informačních systémech*. Praha: Ekopress, 2012. ISBN 978-80-86929-81-1.

(16) ZÝKA,O., *Dimenzionální modelování*.[online]. 2011 [cit. 2013-02-07]. Dostupné z: http://www.profinit.eu/fileadmin/Content/profinit.eu/Academy/NDBI036/05_Dimensionalni_modelovani.pdf

(17) MUNDY, John; THORNWAITE, Warren; KIMBALL, Ralph et al. The Microsoft Data Warehouse Toolkit: With SQL Server 2005 and the Microsoft Business Intelligence Toolset. Wiley Publishing. Canada. 2006.

(18) ISO 3166. *Country Codes*. Geneva: International Organization for Standardization, 2010.

Seznam zkratek

4C – Customer value, Cost to the customer, Convenience, Communication

4P – Product, Price, Placement, Promotion

Aj. – a jiné

Atd. – a tak dále

Apod. a podobně

BI – Business Intelligence

CDC –Change Data Capture

CRM – Customer Relationship Management

CSV – Comma-sepparated value

DS – Datový sklad

DSA – Dočasné úložiště dat

DW – Data Warehouse

ECCD – Extract, Conform, Clean, Deliver

EDW – Enterprise Data Warehouse

ELT – Extract, Load, Transform

ERP - Enteprise Resources Planning

ETL – Extract, Transform, Load

FK – Foreign key (Cizí klíč)

GUI – Graphical User Interface

IBM - International Business Machines Corporation

KPI – Key Performance Indicator

MDX - MultiDimensional eXpressions

MS - Microsoft

MSSQL – Microsoft SQL Server 2012

Např. – Například

NF – Normální forma

ODS – Operational Data Store

OLAP – On-Line Analytical Processing

OLTP – On-Line Transactional Processing

OS – Operační systém

PK – Primární klíč

SK – Surrogate Key

SQL – Structured Query Language

SSIS – SQL Server Integration Services

Tzv. – takzvané/ takzvaně

XLS – Soubor aplikace Excel

XML – Extensible Markup Language

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová (bakalářská) práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že bibliografické údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou (bakalářskou) práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 2.5.2013

……..

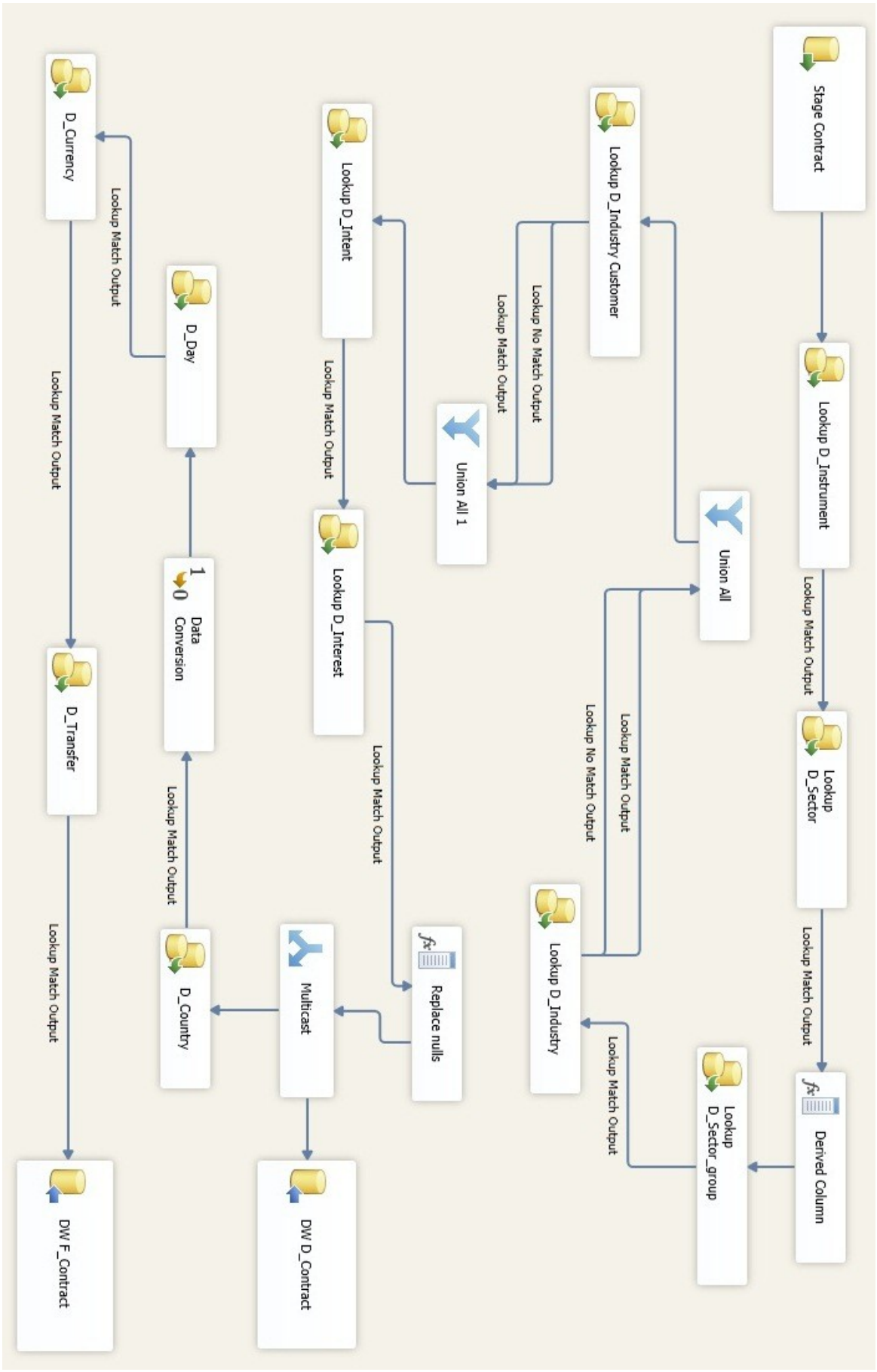
Lukáš Derján

Seznam obrázků

Seznam tabulek

Seznam příloh

Příloha č. 1 – ETL Process tabulek D_Contract a F_Contract



Příloha č. 2 – Mapovací předpis

Column Name	Keep	Transformation	Name	Table
MTRLDATE	ok	Data type Conversion DT_STR to DT_DBDATE	Material date	F
ORGID	ok		Original ID	D
SEPARTID	ok		Separate ID	D
PRODNO	ok		Product ID	D
INSTRUMENT_ID	ok	Lookup D_Instrument	Instrument key	D
COUNTRCD	ok	ISNULL(COUNTRCD) ? "0" : COUNTRCD	Flag international commitment	D
BASECRCD	ok		Flag base currency euro	D
CURNCYCD	ok	Lookup D_Currency	Currency key	F
SECTORCD	ok	Lookup D_Sector + “s” prefix	Sector group key	D
LDGRACNO	ok	ISNULL(LDGRACNO) ? "N/A" : LDGRACNO	Ledger account	D
CREDTBAL	ok	Credtbal vs. debtbal – 1:0	Credit Ballance	D
DEBTBAL	ok	Debtbal vs. credtbal – 1:0	Debit Ballance	D
APPID	ok		Application ID	F
BASKETCD	ok		Flag basket currency	D
BKFORBCD	ok	ISNULL(BKFORBCD) ? "0" : BKFORBCD		D
BRANCHCD	ok	ISNULL(INDUSTRY_KEY_Customer) ? 1 : INDUSTRY_KEY_Customer	Customer Industry key	D
CNTRCTYP	ok		Contract type	F
COLRECNO	ok	ISNULL(COLRECNO) ? "N/A" : COLRECNO	Collateral code	D
COUNTRY	ok	Lookup D_Country	Country key	F
CRDNEWBU				
DBTNEWBU				
NWDRADOW	ok		New Loans	D
OVERDRAW	ok		Overdraw	D
INDUSTRY_ID	ok	Lookup D_Industry, default: 1	Industry key	D
INTENT_ID	ok	Lookup D_Intent	Intent key	D
INTEREST_RATE_ID	ok	Lookup D_Interest_ref_rate	Interest rate key	D
TRANSFER_ID	ok	Lookup D_Transfer	Transfer_Key	F
CREATE_TS		create new	Create_TS	D
rows				
SECTOR_ID	ok	Lookup D_Sector	Sector key	D
AMOUNT_TYPE_KEY	ok	Lookup D_Amount_type	Amount type key	F
		(COLRECNO) == "N/A" ? 0 : 1	Flag collateral record	D
		(BRANCHCD) == "N/A" ? 0 : 1	Flag foreign branch office	D

Příloha č. 3 – Diagram datového skladu